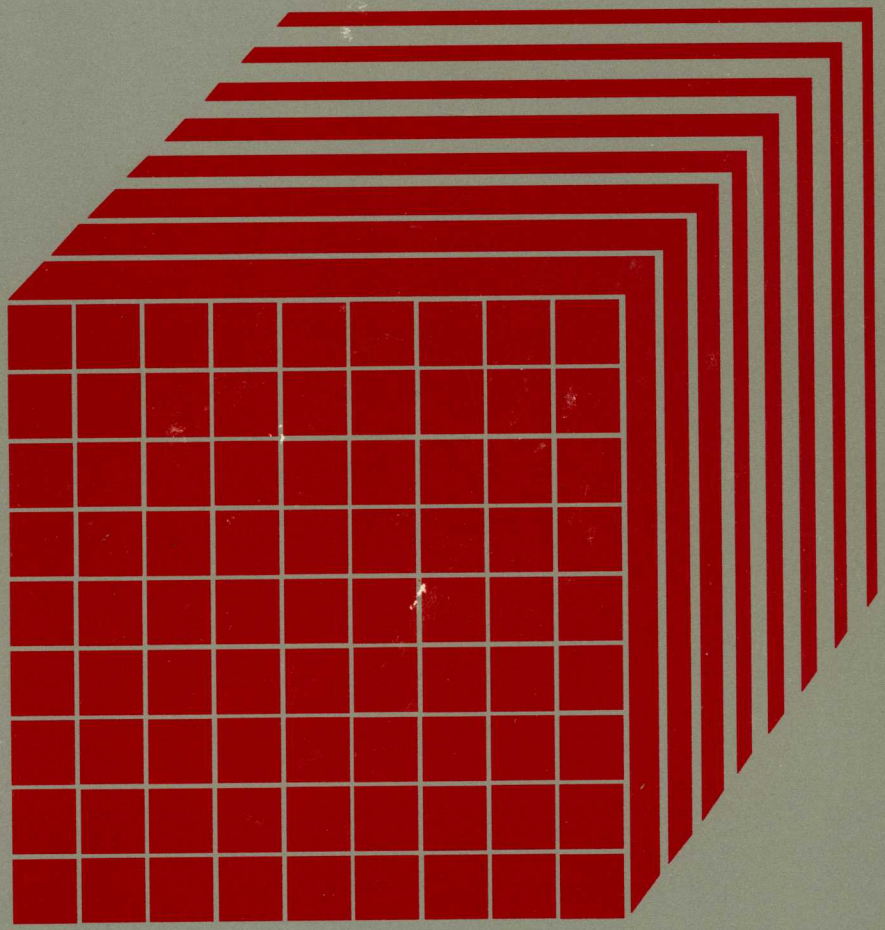
IBM

Virtual Machine/ System Product

Installation Guide

Release 5

SC24-5237-3



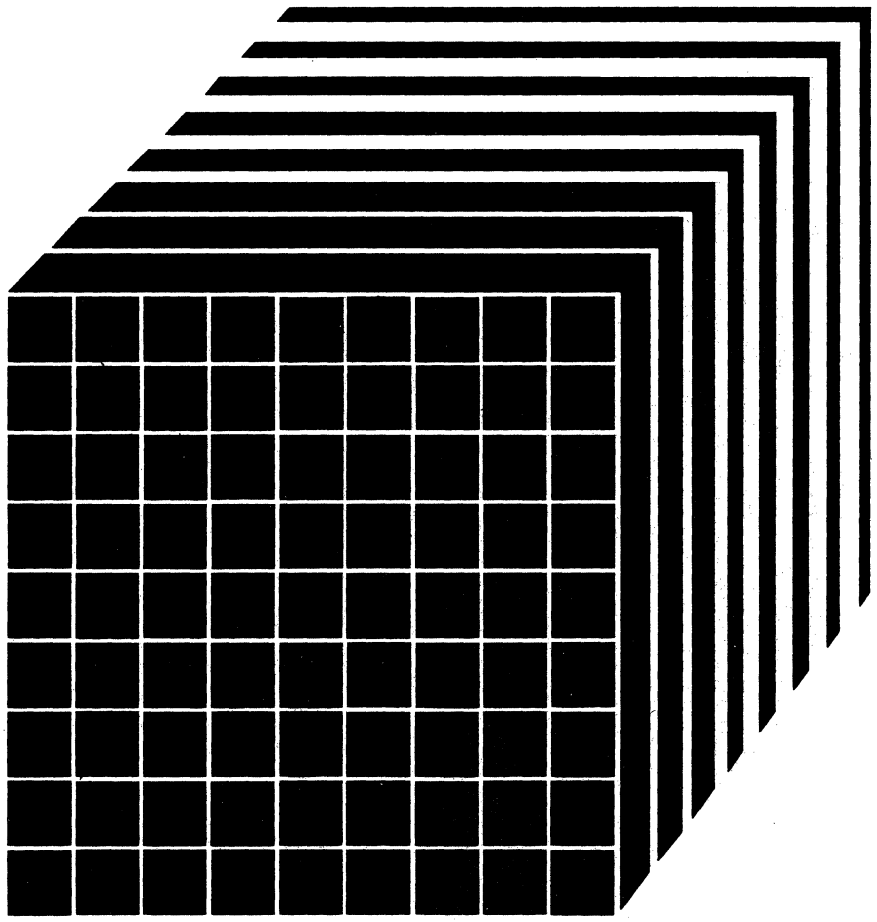
IBM

Virtual Machine/ System Product

Installation Guide

Release 5

SC24-5237-3



Fourth Edition (December 1986)

This edition, SC24-5237-3, is a major revision of SC24-5237-2, and applies to Release 5 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, and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of the major changes, see page X-1.

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

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.

Technical Newsletter

This Newsetter No. SN24-5764
Date December 2, 1988

Base Publication No. SC24-5237-3
File No. S370/4300-34

Previous Newsletters SN24-5761
SN24-5763

IBM Virtual Machine/System Product Installation Guide

This Technical Newsletter, a part of VM/SP Release 5, applies to the *IBM Virtual Machine/System Product Installation Guide* (Program Number 5664-167). It provides additional or replacement pages for your publication. These pages remain in effect until specifically altered. Pages to be added or replaced are:

Title Page, Notices	10-27 to 10-32
v to xviii	X-1 to X-6
8-1 to 8-26	X-21 to X-57
9-1 to 9-40	

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Changes

This Technical Newsletter includes information about the installation of optional feature program products available through the System Offering. For a complete list of changes, see page X-1.

Note: *Please insert this page in your publication to provide a record of changes.*

IBM Corporation, Information Development, Dept. G60, P.O. Box 6, Endicott, N.Y. 13760

Printed in U. S. A.

Technical Newsletter

This Newsetter No. SN24-5761
Date June 12, 1987

Base Publication No. SC24-5237-3
File No. S370/4300-34

Previous Newsletters None

IBM Virtual Machine/System Product Installation Guide

This Technical Newsletter, a part of VM/SP Release 5, applies to the *IBM Virtual Machine/System Product Installation Guide* (Program Number 5664-167). It provides additional or replacement pages for your publication. These pages remain in effect until specifically altered. Pages to be added or replaced are:

Title Page, Notices	10-1 to 10-36
v to xviii	X-1 to X-8
8-1 to 8-32	X-23 to X-59
9-1 to 9-34	

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Changes

This Technical Newsletter includes information about the installation of optional feature program products available through the VM/SP System Offering. For a complete list of changes, see page X-1.

Note: *Please insert this page in your publication to provide a record of changes.*

IBM Corporation, Information Development, Dept. G60, P.O. Box 6, Endicott, N.Y. 13760

Printed in U. S. A.

Preface

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

For a list of the major changes from the previous edition of this book, see page X-1.

For a list of related publications, see page X-15.

This book contains descriptions of the magnetic tapes and direct access storage device (DASD) disk packs required to generate VM/SP. Step-by-step procedures for using them are also included. These descriptions and procedures show you how to:

- Format and allocate the DASD volumes needed to install a system
- Install VM/SP using the Starter System
- Install VM/SP using an existing VM system
- Tailor the new system to meet your specific needs
- Verify that the newly installed CP and CMS components are working properly
- Display product tape files and disk pack layouts
- Install discontiguous saved segments (DCSS)
- Install a new system national language
- Update VM/SP through recommended service procedures
- Use installation and service EXECs and commands.

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 Planning Guide and Reference*, and the *VM/SP System Product Editor User's Guide.*

This book has four parts, plus appendixes:

- "Part 1. VM/SP Installation" provides the step-by-step procedure for installing the CP, CMS, IPCS, GCS, and TSAF components of VM/SP from the VM/SP Product Tape. Two separate procedures are presented:

  - The Starter System procedure is intended for those who are installing VM/SP for the first time. The Starter System procedure is also recommended for current VM users who are installing VM/SP to update their previous VM release.

  - The existing VM system procedure is provided for current VM users who do not have enough available system storage to use the Starter System procedure, or whose present system configuration differs substantially from the sample configuration defined in the sample files provided on the VM/SP Product Tape.

  Part 1 also includes procedures to:

  - Verify that CP and CMS are working properly
  - Install discontiguous saved segments (DCSS)
  - Install a new system national language.

- "Part 2. Optional Feature Program Product Installation" contains information about optional feature program products and how to install them on VM/SP.

- "Part 3. VM/SP Service" describes the procedures and programs used to update VM/SP source code, object code, macro libraries, and text libraries.

- "Part 4. VM/SP Installation/Service EXECs and Commands" describes the EXECs and commands used to install and update VM/SP.

- Appendixes provide information about:

  - Minidisks reserved for the MAINT userid
  - Restricted logon passwords
  - Sample installation and system generation profiles
  - Special options for CP
  - Migrating spool files using SHUTDOWN/WARM IPL
  - Enlarging the CMS nucleus to contain the Y-disk directory
  - Alternate CMS nucleus placement
  - Alternate GCS nucleus placement
  - Sample EXEC procedure for copying VSE macros into a CMS MACLIB
  - Generating and loading the 3704/3705 Control Program
  - Building CP and CMS nuclei using SPGEN
  - Control file identifiers
  - CP/CMS/IPCS module/segment regeneration requirements
  - Servicing Systems Network Architecture (SNA) products.

Contents

Part 1. VM/SP Installation

Part 3. VM/SP Service

Appendixes

Figures

Part 1. VM/SP Installation

Chapter 1. Introduction to VM/SP Installation

Note: If you are installing VM/SP HPO (High Performance Option), use the **VM/SP HPO Installation Guide** *instead of this manual.*

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

THIS BOOK IS YOUR PRIMARY INSTALLATION DOCUMENT. From here, you will be directed to other documentation when necessary.

Pre-Installation Checklist

Do not begin the installation procedure until you complete ALL of 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
- Sample files
- Tools and profiles for installation and system generation
- First level and second level installation
- The format used in this book.

If you are new to VM/SP, you may find these general discussions helpful. If you have previous VM/SP experience, you may 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* may 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 may 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.

How to Organize Your Materials

To install the tapes that you have received, you need to use specific installation documents. You may also want to have other documents available for reference. To make the process easier, organize your materials as shown in the lists in Figure 1-1 on page 1-3. These lists generally correspond to the sequence in which you use the materials during the installation procedure. Keep in mind that:

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

Tapes

Starter System Tape
VM/SP Product Tape
Source feature tape
EREP distribution tape
National language feature tape(s)
Optional feature program product tape(s)
Program update tape (PUT)
.
.
.

Other Installation Documents

VM/SP Program Directory and attachments
Optional feature program product installation documents
.
.
.

Reference Materials

Memo to Licensees
Licensed Program Specifications
VM/SP Planning Guide and Reference
VM/SP Transparent Services Access Facility Reference
VM/SP CP for System Programming
VM System Facilities for Programming
Device Support Facility User's Guide and Reference
VM/SP CP Command Reference
VM/SP System Messages and Codes
VM/SP CMS Command Reference
EREP User's Guide and Reference
.
.
.

Figure 1-1. Organizing Your Materials

VM/SP Overview

VM/SP System Components

VM/SP (Virtual Machine/System Product) is an interactive multiple-access operating system composed 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.

- **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 RSCS (Remote Spooling Communications Subsystem) Version 2 or SNA (Systems Network Architecture) products.

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

Real and Virtual Machines

When you install 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 userid listed in the VM/SP system directory. 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 identifies the device in the virtual machine, just as the real address identifies the device in the real machine. The virtual address may be the same hexadecimal number as the real address, or it may 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 DASD (direct access storage device) disk volumes into small areas called **minidisks**. Minidisks are defined in the VM/SP system directory and are assigned to specific virtual machines. Often called virtual DASD or virtual disks because they simulate real DASD within the virtual machine, minidisks have only virtual addresses. A minidisk that occupies an entire real DASD volume is called a **full-pack minidisk**.

During the installation procedure, you are automatically logged on to the MAINT virtual machine. MAINT is the "system owner," responsible for installing and maintaining the system. Files are loaded to specific MAINT minidisks (see Appendix A, "Minidisks Reserved for the MAINT Userid" on page A-1). For additional information about the MAINT userid, see "A Virtual Machine for Updating VM/SP" on page 11-3.

For a more detailed explanation of the virtual concept, refer to the *VM/SP Introduction*.

Installation Overview

VM/SP installation consists of the following basic processes:

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

These processes require you to issue commands, enter data, 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** of a completely new VM/SP system that does not include any portion of any existing system.

- **Migration** of an existing VM system to the new VM/SP release level.

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 system procedure. Both installation procedures are provided in this book.

The Starter System

The Starter System is an abbreviated VM 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 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 system residence volume (VMSRES), where CP and CMS reside. Therefore, the Starter System must match the device type of the DASD used for VMSRES. The following Starter System versions are available:

- 3330-11
- 3350
- 3375
- 3380
- FBA (3370/3370-2).

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 1-2, 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 before you load any files to them.
- File 2 contains DDR (DASD Dump/Restore), 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.

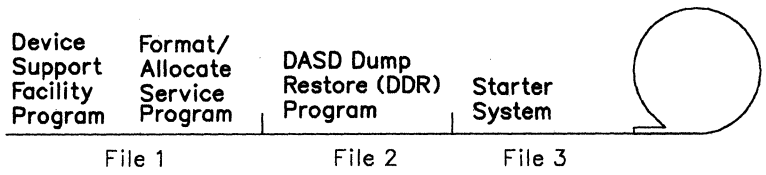


Figure 1-2. VM/SP Starter System Tape File Layout

The VM/SP Product Tape

The VM/SP Product Tape contains:

- Tools and profiles for installation and system generation
- Sample files
- Object code for the CP, CMS, IPCS, GCS, and TSAF components
- HELP files
- Source code for CP, CMS, and IPCS.

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 Figure 1-3. Throughout the rest of this book, the term *product tape* collectively refers to all of the volumes that make up the VM/SP Product Tape.

1600 BPI	6250/38K BPI
Tape 1 File 1: Header File 2: Installation tools File 3: System generation tools File 4: Sample files (DMSNGP, DMKBOX, DMKFCB, RPWLIST) File 5: Sample files (DIRECT, DMKSNT, EXTENTS) File 6: Sample files (DMKRIO, DMKSYS) File 7: CP object code File 8: CMS system code File 9: CMS base code Tape 2 File 1: Header File 2: IPCS object code File 3: HELP files File 4: GCS interface files File 5: GCS object code File 6: TSAF interface files File 7: TSAF object code Tape 3 File 1: Header File 2: CP source code Tape 4 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: Sample files (DMSNGP, DMKBOX, DMKFCB, RPWLIST) File 5: Sample files (DIRECT, DMKSNT, EXTENTS) File 6: Sample files (DMKRIO, DMKSYS) File 7: CP object code File 8: CMS system code File 9: CMS base code File 10: IPCS object code File 11: HELP files File 12: GCS interface files File 13: GCS object code File 14: TSAF interface files File 15: TSAF object code Tape 2 File 1: Header File 2: IPCS source code File 3: CP source code File 4: CMS source code

Figure 1-3. VM/SP Product Tape Volumes File Layout

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.

Sample Files

To help you install your VM/SP system, sample files are shipped with the VM/SP code on the product tape. These files contain information used in the installation process to define various VM/SP system parameters. However, because these files provide only sample information and defaults that may or may not meet your requirements, during the installation procedure you must edit some of the files and tailor the data to define the unique configuration of your system.

System Definition Files

The system definition files are a group of sample files that are essential to the generation of your VM/SP system:

- The **system directory (VMUSERS DIRECT)** identifies the virtual machines (users) permitted to log on to your system. A directory entry for a virtual machine defines the userid of the machine, the password that enables the userid to log on, the virtual input/output device configuration, the virtual processor size, the minidisks assigned or LINKed to the userid, 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 discontiguous saved segments (DCSS).

- 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 and save a CMS nucleus.

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

Other Sample Files

The following files also affect the configuration of your system:

- The **DMKBOX ASSEMBLE** file defines the logo that appears on your screen when you log on your installed VM/SP system. If you want to change the design and/or contents of the logo, you must tailor this file.

- The **RPWLIST DATA** file contains a list of restricted logon passwords. These passwords, which were used in the sample 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 **ADRP** (Auto-Deactivation of Restricted Passwords) uses the RPWLIST DATA file to purge the system of all restricted passwords.

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

  - **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, invokes other EXECs and commands to do most installation and system generation tasks.

SPGEN EXEC This tool directs various system generation and maintenance functions, using the information contained in SPGEN PROFILE. These functions include:

- Creating, verifying, and displaying system profile parameters
- Assembling system files
- Generating CP, CMS, and GCS nuclei
- Receiving and verifying load maps.

SPGEN EXEC is invoked by ITASK, or you can invoke it directly.

SPGEN PROFILE This file identifies the loadlists, control files, minidisk access orders, and other information used by SPGEN EXEC to generate CP, CMS, and GCS nuclei. You may accept the sample values as shipped or tailor the file to meet the requirements of your installation.

SPLOAD EXEC This tool directs the loading of tape files from the product tape and feature tapes, using the information contained in SPLOAD PROFILE. SPLOAD EXEC is invoked 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, source feature tape, and national language feature tape; it also specifies the userid and minidisk address where SPLOAD EXEC loads each tape file.

UTILITY EXEC This tool provides four 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 disk:
  - CP Directory program (DIR)
  - Format/Allocate program (FMT)
  - DASD Dump/Restore program (DDR).
- Writing a backup IPLable CP nucleus to tape.

For additional information about these tools and profiles, see Chapter 15, "EXEC and Command Procedures." PROFILEs are described under the corresponding EXECs.

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**. 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**. At second level, your "console" is a terminal logged on to the first level VM/SP system through a valid userid.

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 these tasks.

When you do a second level installation, you may only have access to 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 userid.

Perhaps the most 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 userid. This first level minidisk (which may 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 may 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.Figure 1-4 shows the relationship between first level and second level operation.

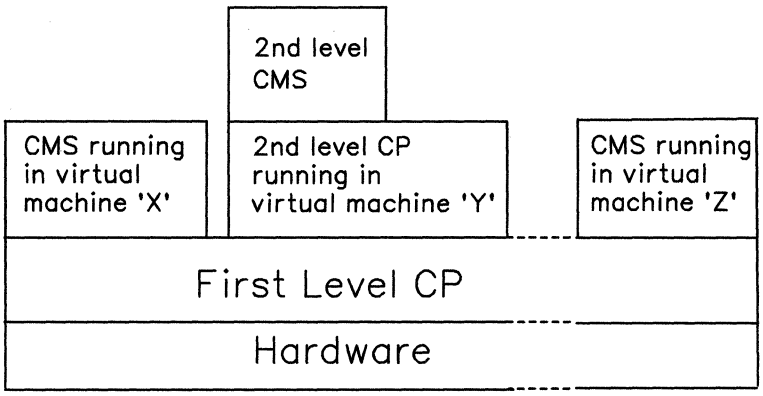


Figure 1-4. Relationship between First Level and Second Level Operation

First level and second level operations are clearly identified in the appropriate steps of the installation procedures in this book.

The Format Used in This Book

The page layout and text formats used to describe procedures in this book are designed to help you identify the following types of information:

- System messages, responses, and prompts

  Note: In the procedures, 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.

This book contains procedures to install and service VM/SP. A procedure usually consists of a series of steps which correspond to natural break points in the procedure. Each step in the procedure may describe one or more operations. Some procedures in this book may assume a higher level of experience, and therefore may not provide as much detail.

In the installation procedures, the number of the step is indicated by a heading at the top of the page. The description of each step contains two sections.

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

The **Procedure** section 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 may receive additional 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 blue shaded background; variables are shown in italics.
. . .	The three vertical dots (which **do not** actually appear on your screen) represent other messages that **may** 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, which is shown in underlined blue lowercase italics, 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 May 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 May Have to Do

Something That You May 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 may want to do. This format is also used to supply some non-procedural additional information, such as error messages for supplemental programs.

ATTN and **ENTER**

These symbols, shown in underlined blue uppercase type, 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.

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 the following table:

Device	ATTN	ENTER
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

Figure 1-5. Console/Terminal Keys That Signal the System

Chapter 2. Planning Your VM/SP Installation

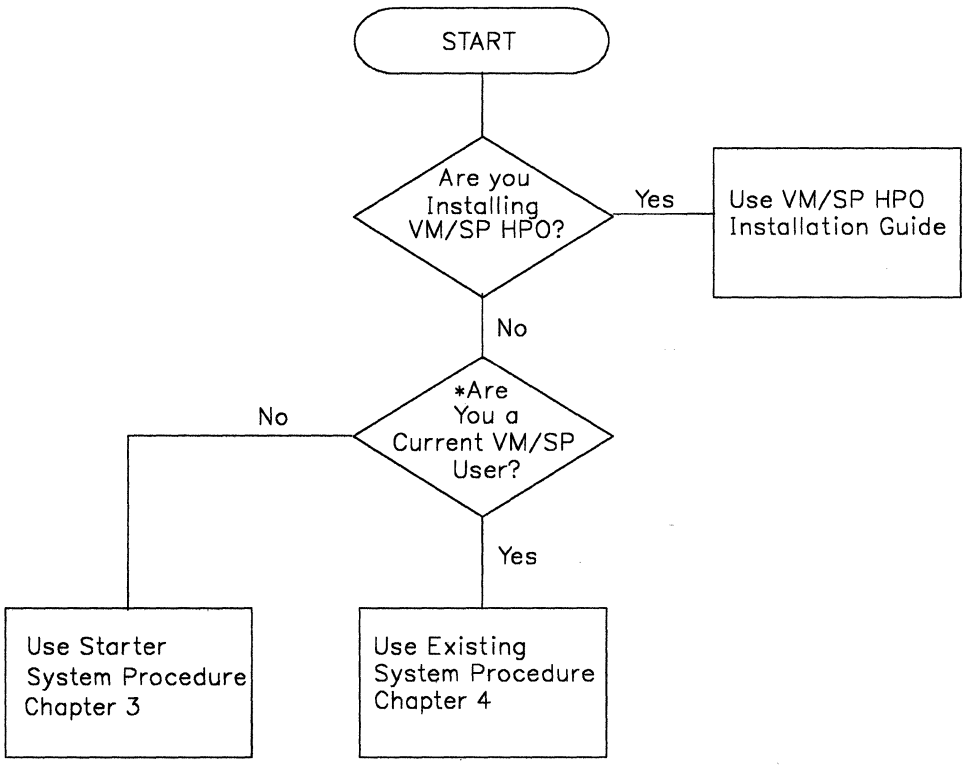
Read this chapter BEFORE you begin your installation.

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
- Migrating spool files
- Identifying your DASD requirements
- Defining your Starter System configuration
- Planning for the 308x Processor Complex
- 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 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.



*If you are a current VM/SP user, you can use the Starter System procedure or the existing system procedure. The existing system procedure assumes a higher VM knowledge level.

Figure 2-1. Selecting the VM/SP Installation Procedure

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 (see Figure 2-1):

- 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 include any portion of any existing VM 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 invoking other tools, such as SPLOAD EXEC and SPGEN EXEC. These tools use installation and nucleus generation instructions supplied by profile files (SPLOAD PROFILE, SPGEN PROFILE, DMSNGP ASSEMBLE). If you accept the sample values supplied in the profiles, you can complete the installation with fewer entries and decisions. If you do not want to accept the sample values, you can modify the profiles 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 system release. To do a migration, the Starter System procedure (Chapter 3) is recommended. If you do not have enough DASD available to use the Starter System (see "Identifying Your DASD Requirements" on page 2-4), or if your system configuration differs substantially from the sample configuration defined in the sample files, you can use the existing VM system procedure (Chapter 4). However, this procedure assumes a greater working knowledge of VM/SP and the installation process.

Migrating Spool Files

If you are doing a migration, and you have spool files that you also wish 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, refer to the *VM/SP CP Command Reference.*

- If you are migrating from VM/SP Release 3 or later, you can use the procedure described in Appendix E, "Migrating Spool Files Using SHUTDOWN/WARM IPL" on page E-1. **This procedure has certain 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 disk layout for CP-owned DASD. You can not use this procedure if you are using the Starter System procedure to migrate VM/SP or if your system residence volume is on a 3380-AE4 or 3380-BE4 DASD.

Identifying Your DASD Requirements

To install the VM/SP base (initial installation or migration), you need the following DASD **volumes** (some physical devices contain more than one volume):

Label	Function
VMSRES	System Residence
VMPK01	Work Volume
VMPK04	Source Code Volume

Notes:

1. *VMPK04 is required only if you plan to load source code.*
2. *If you are using the Starter System procedure, the VMSRES volume must be on a DASD type that matches the Starter System version: 3330-11, 3350, 3375, 3380, or FBA (3370/3370-2).*
3. *If you are using the Starter System procedure, the addresses of your VMSRES and VMPK01 volumes must be X'nn7' or less.*
4. *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 may also want to record the real address of the volume; however, you may 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.

When you IPL the Starter System, 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 may find it helpful to record this information now in your Worksheet:

- If you are installing at **first level**, record your **real** address and device type.
- If you are installing at **second level**, record the **first** level **virtual** address and device type, which function as "real" at second level. You may 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, refer to the *VM/SP Planning Guide and Reference.*

Planning for the Group Control System (GCS)

If you plan to install SNA (Systems Network Architecture) products or RSCS (Remote Spooling Communications Subsystem) Version 2, 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, refer to the *VM/SP Planning Guide and Reference.*

If you plan to install GCS, you must define 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 may 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.

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

- **Calculating your GCS storage requirements**

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

- Tailor the GCS entry in the system directory
- Tailor the GCS entry in the system name table
- Create 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 system directory**

The sample directory (VMUSERS DIRECT) supplied on the product tape contains an entry for a GCS recovery machine (userid 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 restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1). You may also have to change the machine size if you calculated a greater storage requirement.

In addition, you may want to create new directory entries for the userids that will use GCS. Keep in mind that, to use GCS, a userid must LINK to the GCS system disk (MAINT 595). If you do not include a LINK statement for this disk in the user's directory entry, then the user must issue a LINK command. A LINK to the GCS system disk extension (MAINT 59E) is optional, depending on whether or not your GCS applications use this disk.

The *VM/SP Planning Guide and Reference* explains the structure of system 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 create the GCS configuration file.

If you plan to install multiple GCS systems, you must create a new NAMESYS entry for each additional system and select 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 create the GCS configuration file during the installation procedure by invoking GROUP EXEC, which presents a series of panels (screens) that ask you to supply certain information about your GCS system. You may find it helpful to record this information now in your Installation Reference Worksheet (space is provided to define up to three GCS systems).

  Note: If you do not have a full screen display device, you can not use GROUP EXEC, because you can not 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 may 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 system that you want to install. This name must match the SYSNAME parameter in the NAMESYS entry that defines this named system in the DMKSNT file. GROUP EXEC creates a configuration file named *systemname* GROUP, which becomes the blueprint for your virtual machine group.

  Authorized Userids — These are the userids allowed to run in supervisor state and allowed to use GCS functions. For example, these userids have authorization to change common storage.

  System Disk vdev — This is the virtual address of the GCS system disk. MAINT 595 is the default, as defined in SPLOAD PROFILE and in the system 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 userid must have a LINK statement for this disk in the system directory, or the user must issue a LINK command.

  System Disk Ext. vdev — This is the virtual address of the GCS system disk extension. MAINT 59E is the default, as defined in the system directory. This disk is available for use by applications, and LINKs to this disk are optional.

Common Dump Receiver Normally, when a userid's virtual storage is dumped to trace a problem, the dump file is spooled to the userid's virtual reader. However, when you create the configuration file, you can select one authorized userid to receive all GCS virtual storage dumps. If one of your GCS userids is the RSCS virtual machine, you **must** name a common dump receiver, since the RSCS virtual machine cannot process a dump file spooled to its own virtual reader.

Recovery Machine You **must** specify one authorized userid to act as the recovery machine. The sample directory defines a recovery machine called 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 may 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 takes up one control block in common storage.

System ID This is the message that is displayed at the console of a virtual machine operator who IPLs this GCS segment. You can define up to 130 characters.

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

Note: Do not include CMSVSAM and CMSBAM segments here. Instead, follow the procedures described in Chapter 6, "Installing Discontiguous

Saved Segments" on page 6-1, to build these segments 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 userid and node id.

The sample system directory contains an entry for the TSAF virtual machine (userid 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 userids (all userids 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, refer to the *VM/SP Transparent Services Access Facility Reference.*

Tailoring SPGEN PROFILE

SPGEN PROFILE contains various types of information, such as loadlist names, control file names, and minidisk access orders, that the system generation tool (SPGEN EXEC) uses when building system nuclei. During the installation procedure, you can accept the values in SPGEN PROFILE as shipped on the product tape (see "Sample SPGEN PROFILE" on page C-3), or you can tailor the file to satisfy your own requirements. For more information about SPGEN PROFILE, see "SPGEN PROFILE Syntax" on page 15-26 and "SPGEN PROFILE Defaults" on page 15-29.

The first section of SPGEN PROFILE contains information used when generating the CP nucleus, including settings for the following special CP options:

Option	Default Setting
Small CP	NO
CP Fret Trap	NO
Virtual=Real	NO

For additional information about these options, see Appendix D, "Special Options for CP" on page D-1.

Tailoring the System Definition Files

The purpose and structure of these files are described in greater detail in the *VM/SP Planning Guide and Reference.* General tailoring considerations are outlined below. Specific tailoring instructions are provided in the VM/SP installation procedures (Chapter 3 and Chapter 4).

System Directory (VMUSERS DIRECT)

The system directory identifies the virtual machines (users) permitted to log on to your system. As a **system security** measure, the sample system directory defines the logon prevention password **NOLOG** for all userids except MAINT and OPERATOR. You can not log on to any userid 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 called **RPWLIST DATA** (see Appendix B, "Restricted Logon Passwords" on page B-1) that is loaded with the sample files.

A feature called **ADRP** (Auto-Deactivation of Restricted Passwords) is included in the Starter System CP nucleus and the VM/SP CP nucleus. 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 userids to remove the NOLOG condition.

In addition, you may want to:

- Modify certain sample directory entries, such as GCS or TSAFVM.
- Create entries for additional userids, 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 userid. The default userid is OPERATOR; this userid appears in the installation procedure. However, if you want some other userid 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. Since 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 discontiguous saved segments (DCSS). If you plan to install GCS or other products that have segment requirements, you must make sure that the DMKSNT file contains entries for these segments and that the segments are properly defined.

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

Figure 6-1 on page 6-3 shows the suggested segment layout 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, refer to *VM/SP CP for System Programming* for directions.

CMS Nucleus Generation Profile (DMSNGP ASSEMBLE)

DMSNGP defines responses to the system prompts that are generated when you build and save a CMS nucleus. If a response is defined in the DEFNUC statements in this file, the system uses that value, and the corresponding prompt does not appear when you generate the nucleus. If a DEFNUC statement is missing or empty (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 generate the nucleus.

DMSNGP is supplied with most responses pre-defined (see "Sample CMS Nucleus Generation Profile (DMSNGP ASSEMBLE)" on page C-5).

However, you can modify the file to tailor the responses to suit your installation. For more information about DMSNGP and the DEFNUC macro, refer to the *VM/SP Planning Guide and Reference.*

Installing National Languages on Your VM/SP System

VM/SP is shipped with American English as the system national language. This 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 American English.

However, you can order other national languages to install on your system. National languages are distributed on national language feature tapes. The files on a national language feature tape contain translated information.

There are two ways to install a national language:

- As the new system national language (replacing American English).

  Then, when users log on, they receive most messages, see some panels, and enter 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 may vary between national languages, and some messages and panels may still appear in American English.*

  2. *The language of the installation and system generation tools is American English, and can not 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 discontiguous saved segment (DCSS) 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 DCSS, refer to *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				
VMPK04	SPSYS				

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 address of each DASD, tape drive, or graphic device used to restore the Starter System must be X'nn7' or less.

GCS CONFIGURATION FILE

	System #1	System #2	System #3
System Name:			
Authorized Userids:			
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 directory

Chapter 3. Installing VM/SP Using the Starter System

This chapter describes the step-by-step procedure to install VM/SP using the Starter System.

If you are doing an initial installation, whether first level or second level, you must use this procedure.

If you are migrating from a previous VM release, this is the recommended procedure.

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 may receive additional messages, responses, and prompts, or 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.

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 an FBA (fixed-block architecture) device.*

Starter System Installation Procedure Checklist

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

___ At least 768K 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 feature tape for a new system national language, if included with your order

___ At least one additional tape for system backup

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

___ A completed Installation Reference Worksheet.

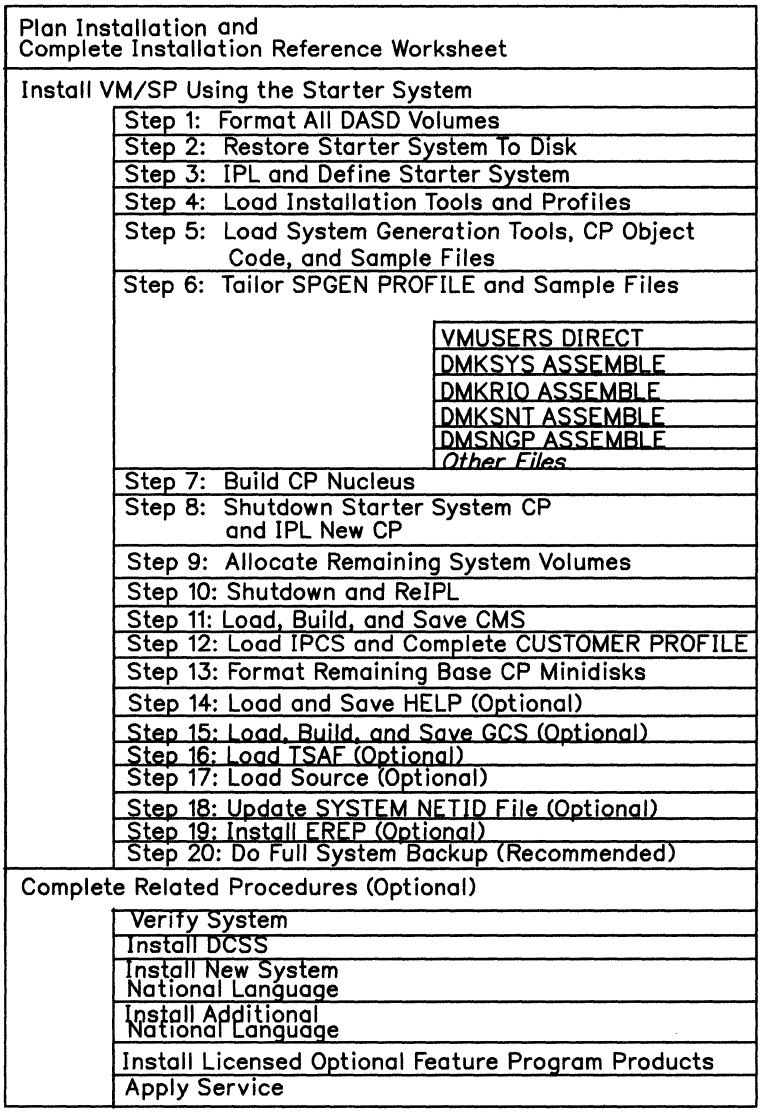


Figure 3-1. Installing VM/SP Using the Starter System

Step 1. Format All 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 VM minidisks
  - Investigating suspected drive problems
  - Checking disk surfaces
  - Assigning alternate tracks
  - Printing a track assignment map
  - Emulating a CKD device on a 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 may be connected with 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, refer to 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 and VMPK01 (and optionally VMPK04). 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). The other volumes are allocated in Step 9.

First Level and Second Level Considerations

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

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

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

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 may 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 the* **ATTN** *or* **ENTER** *function, see Figure 1-5 on page 1-15.*

```
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 may enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The Device Support Facility may 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, which may 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 **3330-11**, **3350**, **3375**, **3380**, or **fb-512** (for the 3370 or 3370-2).

```
ENTER START  PAGE (XXXXX)|CYLINDER (XXX OR XXXX)  OR "LABEL":
```

ENTER

The default start page/cylinder is zero.

```
ENTER END  PAGE (XXXXX)|CYLINDER (XXX OR XXXX):
```

ENTER

The default end page/cylinder depends on the device type that you already entered (see the table below).

Note: The program can tell the difference between models of the same device type (i.e., 3380 and 3380-E4).

Device Type	End Page/Cylinder
3330-11	cyl. 807
3350	cyl. 554
3370	page 69749
3370-2	page 89093
3375	cyl. 958
3380	cyl. 884
3380-E4	cyl. 1769

```
ENTER DEVICE LABEL:
```

spsys

spsys is a temporary label (volid) 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 can not 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.

```
FORMAT STARTED
```

Note: Formatting a large DASD can take up to 30 minutes. 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 may cause the program to issue the message before the formatting is complete.

```
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

If this message indicates **any** errors, stop the installation procedure and contact your Systems Engineer or hardware service personnel. You may have to go back and initialize the DASD volume with the Device Support Facility. Refer to 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 to Disk" on page 3-13.

Procedure, Second Level

- Attach the required DASD to your userid (the system operator may 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 userid and mount the Starter System Tape (the system operator may 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** or **ENTER** function in this procedure, see Figure 1-5 on page 1-15.

ipl ***vdev***

vdev is the virtual address of the tape drive.

Note: After you IPL, if your system appears to be inactive, you may 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 may enter the WAIT (inactive) state. The Device Support Facility may 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, which may 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 **3330-11**, **3350**, **3375**, **3380**, or **fb-512** (for the 3370 or 3370-2).

```
ENTER START  PAGE (XXXXX)|CYLINDER (XXX OR XXXX)  OR "LABEL":
```

ENTER

The default start page/cylinder is zero.

```
ENTER END  PAGE (XXXXX)|CYLINDER (XXX OR XXXX):
```

ENTER

The default end page/cylinder depends on the device type that you already entered (see the table below).

Note: The program can tell the difference between models of the same device type (i.e., 3380 and 3380-E4).

Device Type	End Page/Cylinder
3330-11	cyl. 807
3350	cyl. 554
3370-1	page 69749
3370-2	page 89093
3375	cyl. 958
3380	cyl. 884
3380-E4	cyl. 1769

ENTER DEVICE LABEL:

spsys

spsys is a temporary label (volid) 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 can not 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.

FORMAT STARTED

Note: Formatting a large DASD can take up to 30 minutes. 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 may cause the program to issue the message before the formatting is complete.

```
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

If this message indicates **any** errors, stop the installation procedure and contact your Systems Engineer or hardware service personnel. You may have to go back and initialize the DASD volume with the Device Support Facility. Refer to 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 to Disk" on page 3-13.

Step 2. Restore the Starter System to Disk

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.

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 3-18.

Procedure, First Level

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

  Note: After you IPL, your system may enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The Format/Allocate program may 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 issuing the following commands:

sysprint cons

This command directs system output to the console.

```
ENTER:
```

input ***rdev devtype*** **(rew**

rdev is the real address of the tape drive.

devtype is the device type of the tape drive. Enter **2400**, **2420**, **3410**, **3411**, **3420**, **3422**, **3430**, **3480**, or **8809**.

(rew rewinds the tape without unloading.

```
ENTER:
```

output ***rdev devtype*** **scratch**

rdev is the real address of your system residence volume.

devtype is the device type of the DASD. Enter **3330-11**, **3350**, **3375**, **3380**, or **fb-512** (for the 3370 or 3370-2).

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 tape to disk along with the rest of the volume.

```
ENTER:
```

restore all

```
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
END OF RESTORE

BYTES RESTORED nnnnnnnn
```

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

```
ENTER:
```

ENTER

This response ends the DDR program.

```
END OF JOB
```

Restarting the Restore

If you need to restart the restore:

1. IPL the tape drive **TWO TIMES** to bypass the Device Support Facility and the Format/Allocate program.
2. Go back to the beginning of this step.

3370 DASD Only

The FBA (3370/3370-2) Starter System is supplied with an allocation map for the 3370-2 device. If you restored the FBA Starter System to a 3370 device, you must do a partial allocation to correct the allocation map, which is stored in block 0.

- 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 volume on which the Starter System was just restored.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmsres

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

tdsk 38711 42505
perm 42506 69749
end

```
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 3370 DASD Only

- When you complete the restore, go to "Step 3. IPL and Define the Starter System" on page 3-22.

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 may enter the WAIT (inactive) state. The Format/Allocate program may 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 issuing the following commands:

sysprint cons

This command directs system output to the virtual console.

```
ENTER:
```

input ***vdev devtype*** **(rew**

vdev is the virtual address of the tape drive.

devtype is the device type of the tape drive. Enter **2400**, **2420**, **3410**, **3411**, **3420**, **3422**, **3430**, **3480**, or **8809**.

(rew rewinds the tape without unloading.

```
ENTER:
```

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

vdev is the virtual address of your system residence volume.

devtype is the device type of the DASD. Enter **3330-11**, **3350**, **3375**, **3380**, or **fb-512** (for the 3370 or 3370-2).

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 tape to disk along with the rest of the volume.

ENTER:

restore all

```
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
END OF RESTORE

BYTES RESTORED nnnnnnnn
```

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

ENTER:

ENTER

This response ends the DDR program.

END OF JOB

Restarting the Restore

If you need to restart the restore:

1. IPL the tape drive **TWO TIMES** to bypass the Device Support Facility and the Format/Allocate program.
2. Go back to the beginning of this step.

3370 DASD Only

The FBA (3370/3370-2) Starter System is supplied with an allocation map for the 3370-2 device. If you restored the FBA Starter System to a 3370 device, you must do a partial allocation to correct the allocation map, which is stored in block 0.

- 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 volume on which the Starter System was just restored.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmsres

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

tdsk 38711 42505
perm 42506 69749
end

```
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:
```

ATTN
ENTER

End of 3370 DASD Only

- When you complete the restore, go to "Step 3. IPL and Define the Starter System" on page 3-22.

Step 3. IPL and Define the Starter System

Overview

In this step, you:

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

First and Second Level Considerations

Contention for service by the devices on shared control units may 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 3-23.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-28.

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. Before you perform the system generation, 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 may enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The DDR program may 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 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.

ReIPL 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 reIPLing the Starter System, and the device addresses are not specified correctly, or if the tapes or disks 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 3-26 to initialize the Starter System.

End of ReIPL 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):

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):
```

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):
```

ENTER or ***rdev***

rdev is the address of a graphic device (terminal) other than the system console. 020 is the default.

Warning: Do not define a graphic device at an address with the same control unit (same first two hex digits) as the primary console device.

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

ENTER or ***devtype***

3277 is the default. For a 3290 device, specify **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 n 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

```
.
.
.
```

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 because 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 DMKLNMnnnE messages that appear.

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

DMKCPJ952I nnnnK system storage
```

You are now logged on to the MAINT userid.

For some processor types, *nnnn*K may 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 reIPL.

```
VM/SP Release n STARTER
```

```
ENTER
```

This entry signals the system to set up MAINT's virtual machine by accessing the system minidisk (MAINT 190) and MAINT's A-disk (191) and invoking MAINT's PROFILE EXEC. The PROFILE EXEC issues the TERMINAL MODE VM command (to enable you to use CMS commands) and accesses the MAINT 193 and 295 minidisks.

```
File not found; disk B 295 will not be accessed
```

This message appears because no files have been loaded to the 295 disk yet (the files are loaded in Step 5).

```
R (193)  R/O
Ready;
```

■ Go to "Step 4. Load the Installation Tools and Profiles" on page 3-35.

Step 3, Second Level

Procedure, Second Level

- Obtain support for your second level terminals by issuing 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) to do this installation, issue the following commands:

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 reissue this command.

Note: If you receive the message **DMKCFT006E Invalid device type -**, *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 1m

Your first level system must have **at least** 1 M storage to contain your second level VM/SP system.

ReIPLing Your Second Level System

If you **log off** your **first level** userid 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 reIPL 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 may enter the WAIT (inactive) state. The DDR program may 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 reIPL.

yes or **no**

If you are reIPLing the starter system, and the device addresses are not specified correctly, or if the tapes or disks 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 3-33 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 device type 3262, but used a 1403 to set up your machine in the first level directory, you can not print from your second level machine. Press **ATTN** to return to first level CP and issue the following commands:

```
detach 00e
define 3262 00e
```

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):	
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):	
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):	
ENTER or ***vdev***	***vdev*** is the address of a graphic device (terminal) other than the device that you are using as a console. 020 is the default. **Warning:** Do not define a graphic device at an address with the same control unit (same first two hex digits) as the primary console device.
ENTER DEVICE TYPE (3277,3278,3066):	
ENTER or ***devtype***	3277 is the default. For a 3290 device, specify **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 n 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 VMPK01 not mounted
```

You receive this message because VMPK01 is listed in the listed in the SYSOWN macro in the Starter System CP nucleus. 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 DMKLNMnnnE messages that appear.

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

DMKCPJ952I nnnnK system storage

.
.
.

FORMATTING ERROR RECORDING AREA
```

You are now logged on to the MAINT userid.

For some processor types, *nnnn*K may be less than your actual system storage.

This message appears only during the initial IPL.

Step 3, Second Level

```
DMKCPI966I Initialization complete
```

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

```
VM/SP Release n STARTER
```

ENTER

This entry signals the system to set up MAINT's virtual machine by accessing the system minidisk (MAINT 190) and MAINT's A-disk (191) and invoking MAINT's PROFILE EXEC. The PROFILE EXEC issues the TERMINAL MODE VM command (to enable you to use CMS commands) and accesses the MAINT 193 and 295 minidisks.

```
File not found; disk B 295 will not be accessed
```

This message appears because no files have been loaded to the 295 disk yet (the files are loaded in Step 5).

```
R (193)  R/O
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 issuing the following command to **disconnect** your **first level** userid:

%cp disconn

To restart, log on again to your first level userid and issue the following commands:

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 3-35.

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 tool invokes other EXECs and 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 SPLOAD PROFILE. It is invoked by ITASK, or you may invoke it independently.

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

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

SPGEN PROFILE This file identifies the loadlists, control files, minidisk access orders, and other information that SPGEN EXEC (loaded in Step 5) uses to build system nuclei.

INSTFPP EXEC This is a tool for loading optional feature program products from optional feature program product tapes.

In this step, you use the VMFPLC2 command to load the INSTALL TOOLS tape file from the product tape to the MAINT 191 minidisk. For more information about the VMFPLC2 command, see page 15-72.

First and Second Level Considerations

To do this step at **first level**, go to "Procedure, First Level" on page 3-36.

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

Step 4, First Level

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 on the 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 issuing 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. 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 from virtual address 181 and loads it to the A-disk (in this case, MAINT 191).

Tailoring SPLOAD PROFILE

After you load the installation tools, if you want to change the default userids and minidisk addresses where other tape files are loaded from the product tape, use the System Product Editor (XEDIT) to tailor SPLOAD PROFILE. See "Sample SPLOAD PROFILE" on page C-1.

For instructions on how to use XEDIT, see the *VM/SP System Product Editor User's Guide* or the *VM/SP CMS Primer*.

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.

If you change SPLOAD PROFILE, you may not be able to use the ITASK EXEC installation tool, because ITASK expects to find certain information on specific minidisks. 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, CP Object Code, and Sample Files" on page 3-40.

Step 4, Second Level

Procedure, Second Level

- Mount the first product tape volume on a tape drive attached to your **first level** userid, as defined in Step 1 (the system operator may have to do this for you).

- Attach the tape drive to the MAINT **second level** virtual machine by issuing the following commands:

vary on *vdev*

vdev is the **virtual** address of the tape drive attached to your **first level** userid, 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. 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 from virtual address 181 and loads it to the A-disk (in this case, MAINT 191).

Tailoring SPLOAD PROFILE

After you load the installation tools, if you want to change the default userid and minidisk addresses where other tape files are loaded from the product tape, use the System Product Editor (XEDIT) to tailor SPLOAD PROFILE. See "Sample SPLOAD PROFILE" on page C-1.

For instructions on how to use XEDIT, see the *VM/SP System Product Editor User's Guide* or the *VM/SP CMS Primer*.

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.

If you change SPLOAD PROFILE, you may not be able to use the ITASK EXEC installation tool, because ITASK expects to find certain information on specific minidisks. 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, CP Object Code, and Sample Files" on page 3-40.

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

Overview

Now that you have loaded 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 page 15-12.

In this step, you invoke ITASK with the LOAD CP operands. ITASK invokes SPLOAD EXEC to load the following tape files from the product tape:

- SYSGEN TOOLS to MAINT 193.

  This tape file contains system generation tools, including:

  - SPGEN EXEC
  - UTILITY EXEC
  - SNTMAP EXEC
  - DISKMAP EXEC
  - DIRECGEN EXEC
  - $MSG4I$ EXEC
  - $DASD$ CONSTS

- 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 contains:

  - DMSNGP SAMPLE
  - DMKBOX ASSEMBLE
  - DMKFCB ASSEMBLE
  - RPWLIST DATA

- DASDMODL SAMPLES to MAINT 295.

  This tape file contains:

  - DIRECT *devtype[-mm]*
  - DMKSNT *devtype[-mm]*
  - EXTENTS *devtype[-mm]*

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

- DASDTYPE SAMPLES to MAINT 295.

  This tape file contains:

  - DMKRIO *devtype*
  - DMKSYS *devtype*

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

The product tape contains DASDMODL SAMPLES and DASDTYPE SAMPLES tape files for all of the DASD types and models for which Starter System versions are available. Under the direction of ITASK, SPLOAD EXEC selects and loads the correct DASDMODL SAMPLES and DASDTYPE SAMPLES for the DASD type and model of your VMSRES device, then renames the sample files as follows:

Old Name	**New Name**
DIRECT *devtype[-mm]*	VMUSERS DIRECT
DMKSNT *devtype[-mm]*	DMKSNT ASSEMBLE
EXTENTS *devtype[-mm]*	DIRECGEN EXTENTS
DMKRIO *devtype*	DMKRIO ASSEMBLE
DMKSYS *devtype*	DMKSYS ASSEMBLE
DMSNGP SAMPLE	DMSNGP ASSEMBLE

Step 5

Procedure

Warning: To do the load, ITASK EXEC issues the COPYFILE command with the REPLACE option. This command overwrites any files already on the disk 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 may want to rename the files that you tailored before you invoke ITASK.

- Invoke ITASK EXEC with the LOAD CP operands:

```
itask load cp
```

Note: There is a pause here. SPLOAD EXEC must format the MAINT 194 and 295 minidisks before it can proceed with the load.

```
DMSWSL409I Loading SYSGEN TOOLS to MAINT 193
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:
DMKSNT  ASSEMBLE
DMKRIO  ASSEMBLE
DMKSYS  ASSEMBLE
DMKBOX  ASSEMBLE
DMKFCB  ASSEMBLE
DMSNGP  ASSEMBLE
VMUSERS DIRECT
```

- Go to "Step 6. Tailor SPGEN PROFILE and the Sample Files" on page 3-43.

Step 6. Tailor SPGEN PROFILE and the Sample Files

Overview

As shipped on the product tape, SPGEN PROFILE and the sample files contain sample information and default parameters. 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.

SPGEN PROFILE, which you loaded from the product tape to the MAINT 191 minidisk in Step 4, contains information that the system generation tool (SPGEN EXEC) uses when building system nuclei.

The sample files, which you loaded from the product tape to the MAINT 295 and MAINT 194 minidisks in Step 5, contain the following information:

- The system directory (VMUSERS DIRECT) identifies the virtual machines (userids) 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 discontiguous saved segments (DCSS).
- 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 and save a CMS nucleus.
- The DMKBOX ASSEMBLE file defines the logo that appears on your screen when you log on your system.
- Other sample files (DMKPIA, DMKPIB, DMKUCB, DMKUCC, DMKUCS) define the printer universal character set and the font offset buffer.

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 UTILITY EXEC to print copies of the following files:

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

Procedures

Note: This step provides only guidance and suggestions. For additional information, refer to the **VM/SP Planning Guide and Reference**.

Restarting This Step

If you have to restart this step, make sure that you have read/write access to the MAINT 295 minidisk, where most of the sample files are located. Issue the following command:

access 295 b

Printing the System Definition Files

To print the system definition files, invoke UTILITY EXEC with the PRSAMPLE operand:

utility prsample

The files are spooled to your system printer.

Notes:

1. *If you modified SPLOAD PROFILE and loaded the system definition files to a non-default minidisk, access that minidisk before you invoke UTILITY EXEC.*

2. *If the printer has been turned off, turn the power on, issue the* **vary online** ***cuu*** *command (where* ***cuu*** *is the address of your printer) to identify the device to CP, and then define the printer parameters as indicated below.*

3. *If the print operation fails, or if the printer has just been turned on, the printer parameters may not be defined correctly. Issue the DRAIN command to stop spooling operations on the device. Issue the proper LOADBUF command(s) to define the printer. Issue the START command to restart the device and then invoke UTILITY EXEC. For more information about using the DRAIN, LOADBUF, and START commands, refer to the* **VM/SP CP Command Reference**.

Using the System Product Editor to Tailor the Files

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

To examine or modify a file, invoke XEDIT:

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

SPGEN Considerations

- Tailor SPGEN PROFILE only if you want to change the supplied nucleus generation information. If you accept the defaults, you do not need to tailor this file.

  For information about the contents of SPGEN PROFILE, see "SPGEN PROFILE Syntax" on page 15-26, and "SPGEN PROFILE Defaults" on page 15-29.

  See Appendix D, "Special Options for CP" on page D-1, for more information about the following CP options:

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

Directory Considerations

- Tailor the sample VMUSERS DIRECT file to define your virtual machine (user) configuration.

  Consider the following changes:

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

    Each userid in the system is protected from unauthorized access by a logon password. This logon password is defined in the userid entry in the system directory.

    As a system security measure, the sample system directory defines the logon prevention password NOLOG for all userids except

MAINT and OPERATOR. You can not log on to any userid 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 ADRP (Auto-Deactivation of Restricted Passwords). The new CP nucleus that you build in Step 7 also contains this feature. Under ADRP, issuing 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 userids to remove the NOLOG condition.

For a list of the restricted passwords contained in the RPWLIST DATA file, see Appendix B, "Restricted Logon Passwords" on page B-1.

- **Using Mixed DASD**

  The sample 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 may want to load and print the sample directory for the other DASD type. On the product tape, look for the DIRECT *devtype[-mm]* file.

- **Modifying the GCS Userid**

  If you are installing the GCS component, you may want to change the specification of the sample GCS recovery machine (userid GCS).

- **Modifying the TSAFVM Userid**

  If you are installing the TSAF component, you may want to change the specification of the sample TSAF virtual machine (userid TSAFVM). Remember that all userids in all systems in the TSAF collection must be unique.

- **Adding Userids**

  If you plan to install optional components (such as GCS) or program products (such as RSCS) that require system userids not provided in the sample directory, you should create directory entries for them

now using the format described in the *VM/SP Planning Guide and Reference.*

- **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. *VM/SP CP for System Programming* provides a procedure for user class restructure.

- **Changing Minidisk Read/Write Passwords**

  Sample read/write passwords are provided for the minidisks defined for each userid 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.

- **Defining Minidisks for GCS and TSAF Source Code**

  If you received the source feature tape with your order, and you plan to load GCS or TSAF source code, these minidisks are not defined in the sample directory. The default minidisks specified in SPLOAD PROFILE are:

  - MAINT 597 for GCS
  - MAINT 493 for TSAF.

  For each minidisk that you need, add an MDISK statement under the MAINT userid. For information about the size of the minidisk, see Figure A-2 on page A-3. In the sample directory, source minidisks are defined on VMPK04. For information about the format of the MDISK statement, refer to the *VM/SP Planning Guide and Reference.*

- **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 change the following directory statements to reflect the **real** address of your VMSRES volume:

  ```
  DIRECTORY 123
  MAINT MDISK 123
  ```

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

  diskmap vmusers direct

  DISKMAP produces a file called VMUSERS DISKMAP, which is placed on your A-disk. Use XEDIT to examine the file. For more information about DISKMAP EXEC, see page 15-9.

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

  Warning: Before you issue 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 ASSEMBLE file to define your system residence configuration.

  Consider the following changes:

  - **Allocating Space for Paging**

    The Starter System and the sample files do not provide space for paging. TEMP space is used for paging until you define paging space. Refer to the *VM/SP Planning Guide and Reference* for information about calculating paging space requirements and allocating space for paging in the SYSOWN macro.

  - **Restructuring User Classes**

    If you plan to restructure the supplied user classes, you may need to change the user class definitions in the SYSFCN macro. Refer to 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*. Refer to the *VM/SP Planning Guide and Reference* for the proper format for this entry.

- **Installing an AP or MP System**

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

The tailored DMKSYS ASSEMBLE file is assembled in Step 7.

DMKRIO Considerations

- Tailor the sample DMKRIO ASSEMBLE file to define the configuration of your system input/output devices. Refer to the *VM/SP Planning Guide and Reference* for more information.

  Make sure that the address of your console (or, at second level, the terminal that you are using as a console) matches the console or alternate console address specified in the RIOGEN statement in the CONSOLE section of this file.

The tailored DMKRIO ASSEMBLE file is assembled in Step 7.

DMKSNT Considerations

- Tailor the sample DMKSNT ASSEMBLE file to define the configuration of named saved systems and discontiguous saved segments (DCSS) in virtual storage.

  Warning: Make sure that your changes do not cause overlay of other products on your DASD. Also make sure that no segment overlays another segment in virtual storage unless the segments are NEVER used at the same time. Figure 6-1 on page 6-3 shows the layout of 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 discontiguous saved segments on the volume may be necessary. For reference, you may want to load and print the sample DMKSNT file for the other DASD type. On the product tape, look for the DMKSNT *devtype[-mm]* file.

- **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 saved segments, you should add the DMKSNT entries now. Otherwise, you will have to reassemble DMKSNT and recreate the CP nucleus.

The tailored DMKSNT ASSEMBLE file is assembled in Step 7.

- 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.

- To verify your changes, you should invoke SNTMAP EXEC, an SNT mapping tool that resides on the MAINT 193 minidisk:

  sntmap dmksnt

  SNTMAP produces two output files on your A-disk:

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

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

  For more information about SNTMAP EXEC, see page 15-21.

DMSNGP Considerations

- Tailor the sample DMSNGP ASSEMBLE file to define your CMS configuration.

  Consider the following changes to the DEFNUC macro:

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

- 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
3330	071 cyl
3350	033 cyl
3370	29184 blk
3375	050 cyl
3380	031 cyl

- You may 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.

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.

Other Considerations

- If you want to change the supplied forms control buffer macros, tailor the sample DMKFCB ASSEMBLE file. Refer to *VM/SP CP for System Programming* for directions.
- If you want to change the design or contents of the logo that appears when you log on to the system, tailor the sample DMKBOX ASSEMBLE file.
- If this is a migration, and you have defined new CP commands, refer to *VM/SP CP for System Programming* 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

Refer to the module prologues 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 3-54.

Step 7. Build the New CP Nucleus

Overview

To build your new CP nucleus, you invoke ITASK EXEC with the BUILD CP operands.

ITASK:

- Accesses the MAINT 295 minidisk, where the system definition files are located.
- Invokes the SPGEN ASSEMBLE procedure to assemble the following sample files:
  - DMKBOX ASSEMBLE
  - DMKFCB ASSEMBLE
  - DMKSNT ASSEMBLE
  - DMKSYS ASSEMBLE
  - DMKRIO ASSEMBLE.
- Invokes the SPGEN BUILD procedure to build the CP nucleus using the system definition files and the information contained in SPGEN PROFILE.

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

- Invoke ITASK EXEC with the BUILD CP operands:

itask build cp

```
295 replaces A (191)
DMSUPD181E No update files were found
ASMBLING DMKBOX

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKBOX TEXT A1 CREATED
```

.
.
.

Note: If any statements are flagged during the assembly procedure, the CP nucleus is NOT built. See "Assembly Errors" on page 3-57.

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

Note: If you do not receive the **NUCLEUS LOADED** *message, or if you receive a non-zero return code, see "Load Errors" on page 3-58.*

The **NUCLEUS LOADED** message gives the starting and ending cylinders/blocks for the nucleus. If you tailored the system directory in Step 6 and added minidisks or changed minidisk allocations, you should verify that this nucleus has not expanded into another allocated area such as TEMP, WARM START, etc., since unpredictable results may occur.

An easy way to check this is to look at VMUSERS DISKMAP (which you created after you tailored the directory) and make sure that **LAST CYL/BLK USED=***nnn*, as displayed above, does not exceed the $CPNUC$ ending cylinder/block as defined in the directory. To look at this file, you must first issue the command: **#cp ipl 190**.

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

This is the message that you receive if the nucleus loaded correctly. However, on many processors this message does not display automatically. If the PSW message does not display within about one minute, issue **#cp display psw**.

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

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 disk, issue the following commands:

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

After **VM READ** appears in the corner of your screen, issue:

```
ENTER
access 194 c
query rdr
```

```
MAINT spoolid M PUN nnnnnnnn 001 NONE STANDARD
```

```
receive spoolid cpnuc map c
```

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

```
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 3-62.

Assembly Errors

- If any DMKxxx ASSEMBLE file fails to assemble, you may have made errors when tailoring the file in Step 6. To correct the errors:

1. Examine the flagged statement(s).
2. Correct the statement(s) using the System Product Editor (XEDIT).
3. Re-assemble the file:

itask assemble dmk*xxx*

dmk*xxx* is the name of the file to be assembled.

4. Repeat this sequence until the file assembles correctly.
5. Go back to the beginning of Step 7, on page 3-55, and reissue the following command:

itask build cp

Reissuing this command ensures that all of the sample files are assembled correctly before the nucleus is built; during the previous attempted CP build, the assembly procedure may have stopped before all of the files were assembled.

Load Errors

- If the new nucleus fails to load:

  1. Inspect the CP load map and the virtual PSW. A loader error may be indicated in the load map or in the PSW. Refer to *VM/SP System Messages and Codes* for a list of the loader wait state codes.
  2. Correct the error.
  3. Issue the following command:

itask build cp noassem

- If you shut down following a nucleus loading error and can not 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 **3330-11**, **3350**, **3375**, **3380**, or **fb-512** (for the 3370 or 3370-2).

```
ENTER START  PAGE (XXXXX)|CYLINDER (XXX OR XXXX)  OR "LABEL":
```

nnnnn — Enter the start page/cylinder according to your DASD type. See Figure 3-2 on page 3-59.

Warning: Do not press ENTER before entering the start page/cylinder. If you do, you will begin formatting at page/cylinder zero and wipe out your previous work.

```
ENTER END  PAGE (XXXXX)|CYLINDER (XXX OR XXXX):
```

nnnnn

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

Warning: Do not press ENTER before entering the end page/cylinder. If you do, you will format up to the last page/cylinder and wipe out your previous work.

Device Type	WARM START Start Cyl/Blk	WARM START End Cyl/Blk	CHECKPOINT Start Cyl/Blk	CHECKPOINT End Cyl/Blk
3330-11	34	36	325	326
3350	17	18	200	200
3375	24	25	325	325
3380	17	18	271	271
FBA	13480	14503	187224	187735

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

```
ENTER DEVICE LABEL:
```

vmsres

```
FORMAT STARTED

FORMAT DONE
000 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.
6. Define your input and output devices:

sysprint cons

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

input *cuu devtype*

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

devtype is the device type of your tape drive.

output *cuu devtype* scratch

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

devtype is the device type of your VMSRES DASD.

7. Restore the Starter System:

restore *allbeg allend*

This command restores the Starter System allocation. The values of ***allbeg*** and ***allend*** are determine by your VMSRES DASD type. See Figure 3-3.

restore *nucbeg nucend*

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

restore *dirbeg dirend*

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

Device Type	allbeg	allend	nucbeg	nucend	dirbeg	dirend
3330-11	000	000	001	010	469	473
3350	000	000	001	005	308	310
3375	000	000	001	007	535	538
3380	000	000	001	005	500	501
FBA	00000	00015	000016	004119	248472	250295

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

8. (Second level only) Obtain support for your second level terminals (see page 3-28.)

9. IPL your VMSRES volume.

10. Rewrite the system 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 3-55 and reissue the **itask build cp** command to reassemble all of the files and rebuild the nucleus.

    If you DO NOT find any errors in the system definition files, issue the following command to rebuild the nucleus:

itask build cp 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 userid specified by the SYSOPER keyword in the SYSOPR macro in the DMKSYS file. The default userid is OPERATOR.

While you are logged on as OPERATOR, format the OPERATOR 191 minidisk. You can not 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. When you have logged on as MAINT, you can use UTILITY EXEC to save an IPLable copy of your CP nucleus on tape for recovery purposes. You may 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 3-63.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-67.

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 n, 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 may 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 userid.

DMKCPJ952I *nnnn*K system storage

Based on your processor type, *nnnn*K may be less than your actual system storage.

.
.
.

DMKCPI966I Initialization complete

define storage 16m

ipl 190

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

VM/SP Release *n* STARTER

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

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 disk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the disk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, refer to 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

Formatting disk A

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

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

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

- Log on to the MAINT userid by doing ONE of the following:
  - Issue the **disconn** command to disconnect OPERATOR, then log on to the console as MAINT.

    OR

  - Issue 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

When you have logged on as MAINT, you can save a copy of your CP nucleus on tape for recovery purposes. Invoke 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 the system directory (see page 3-48) and in DMKSYS (see page 3-49).*

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, refer to 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 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 3-71.

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'
```

- 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 n, 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 may 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.

Step 8, Second Level

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

You are now logged on to the OPERATOR userid.

```
DMKCPJ952I nnnnK system storage
```

Based on your processor type, *nnnn*K may be less than your actual system storage.

```
.
.
.
```

```
DMKCPI966I Initialization complete
```

define storage 16m

ipl 190

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

```
VM/SP Release n STARTER
```

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

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 disk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the disk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, refer to 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 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 userid by doing ONE of the following:
  - Issue 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. Issue 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 graphics 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. Issue the following command **from the second terminal** to identify the terminal to your system. In this command, ***userid*** is your **first level** userid. This completes the link between the graphics 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

Once you have logged on as MAINT, you can save a copy of your CP nucleus on tape for recovery purposes. Invoke 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 the system directory (see page 3-48) and in DMKSYS (see page 3-49).*

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, refer to 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.

- To continue the installation procedure, go to "Step 9. Allocate the Remaining System Volumes" on page 3-71.

Step 9. Allocate the Remaining System Volumes

Overview

Before you load the remaining product tape files to minidisks, 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 ITASK EXEC with the ALLOCATE operand.

ITASK:

- Looks at the $ALLOC$ userid in the system directory to obtain the permanent system volume labels.
- Prompts you for the addresses of the volumes that you want to allocate.
- Invokes 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 system 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 3-72.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-74.

Step 9, First Level

Procedure, First Level

Note: Do this step while logged on to the MAINT userid.

- Mark your DASD volumes available to CP (they were marked unavailable by the shutdown in Step 8). Issue 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.

- Invoke ITASK EXEC with the ALLOCATE operand:

itask allocate

```
WHAT IS THE REAL ADDRESS OF YOUR volid VOLUME?
TYPE: REAL ADDRESS OR SKIP
```

You are prompted for each DASD volume listed in the SYSOWN macro in the DMKSYS file. You may 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.

```
DASD rdev ATTACH TO MAINT nnn
```

This message appears only if you entered an address at the previous prompt.

.
.
.

```
PUN FILE nnn TO MAINT  COPY nnn NOHOLD
```

This message appears only once, after you have completed the previous sequence for each volume in the SYSOWN list.

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

```
FORMAT
```

.
.
.

Note: The system displays the format and allocate sequences for each volume that you are allocating (the actual 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 $ALLOC$ userid in the system directory.

```
ALLOCATION RESULTS
PERM      nnnnn   nnnnn
TEMP      nnnnn   nnnnn
PERM      nnnnn   nnnnn
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 may vary.

■ Go to "Step 10. Shut Down and ReIPL" on page 3-76.

Step 9, Second Level

Procedure, Second Level

Note: Do this step while logged on to the MAINT userid.

- Mark your DASD volumes available to CP (they were marked unavailable by the shutdown in Step 8). Issue 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.

- Invoke ITASK EXEC with the ALLOCATE operand:

itask allocate

```
WHAT IS THE REAL ADDRESS OF YOUR volid VOLUME?
TYPE: REAL ADDRESS OR SKIP
```

You are prompted for each DASD volume listed in the SYSOWN macro in the DMKSYS file. You may 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.

Remember that the "real" address at second level is the **first** level **virtual** address of the volume.

vdev or **skip**

Enter the "real" address, ***vdev***, if you want to allocate this volume.

Enter **skip** if you do not want to allocate this volume.

```
DASD vdev ATTACH TO MAINT nnn
```

This message appears only if you entered an address at the previous prompt.

.
.
.

```
PUN FILE nnn TO MAINT  COPY nnn NOHOLD
```

This message appears only once, after you have completed the previous sequence for each volume in the SYSOWN list.

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

FORMAT

.
.
.
```

Note: The system displays the format and allocate sequences for each volume that you are allocating (the actual 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 $ALLOC$ userid in the system directory.

```
ALLOCATION RESULTS
PERM      nnnnn   nnnnn
TEMP      nnnnn   nnnnn
PERM      nnnnn   nnnnn
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 may vary.

■ Go to "Step 10. Shut Down and ReIPL" on page 3-76.

Step 10. Shut Down and ReIPL

Overview

At this point, you do a shutdown and reIPL to designate as "CP-owned" the system volumes that were allocated and named in Step 9.

The reIPL automatically logs you on to the userid specified by the SYSOPER keyword in the SYSOPR macro in the DMKSYS file. The default userid is OPERATOR.

After the reIPL, you must log on again as MAINT to continue the installation procedure.

Procedure

Note: After the first shutdown (Step 8), if you logged on to the MAINT userid at a second terminal, go back now to your primary device and do this shutdown using the OPERATOR userid. After the shutdown and re-IPL are complete, OPERATOR is automatically logged on to the primary device, and you can log on to your second terminal again as MAINT to continue the installation 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

DMKCPJ952I nnnnK system storage

.
.
.

DMKCPI966I Initialization complete
```

You are now logged on to the OPERATOR userid.

Based on your processor type, *nnnn*K may be less than your actual system storage.

- Log on to the MAINT userid by doing ONE of the following:
  - Issue the **#cp disconn** command to disconnect the OPERATOR userid, then log on to the same device as MAINT.

    OR

  - Log on as MAINT at your previously defined second terminal.
- Go to "Step 11. Load, Build, and Save CMS" on page 3-78.

Step 11. Load, Build, and Save CMS

Overview

To load, build, and save CMS, you invoke ITASK EXEC with the LOAD CMS operands.

ITASK:

- Invokes SPLOAD EXEC to load the following tape files from the product tape:
  - CMS SYSTEM to the MAINT 190 minidisk.

    This tape file contains EXECs and MODULEs (commands) used by the general CMS user.
  - CMS BASE to the MAINT 193 minidisk.

    This tape file contains TEXT files and other items used primarily by system programmers.
- Invokes ASMGEND EXEC to build the system assembler and create the auxiliary directory.
- Invokes the SPGEN ASSEMBLE procedure to assemble the CMS nucleus generation profile (DMSNGP ASSEMBLE).
- Invokes the SPGEN BUILD procedure to build and save the CMS nucleus using the information contained in SPGEN PROFILE and DMSNGP ASSEMBLE.

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.

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 3-85.

Procedure, First Level

Note: Do this step while logged on to the MAINT userid.

- 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. SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

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

- Invoke ITASK EXEC with the LOAD CMS operands:

itask load cms

```
DMSWSL409I Loading CMS SYSTEM to MAINT 190
DMSWSL409I Loading CMS BASE to MAINT 193

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 may happen if you modified SPLOAD PROFILE and changed the target disk for the CMS SYSTEM tape file from MAINT 190 to some other disk. To correct this situation, access the disk where the CMS SYSTEM files are located, issue the* **asmgend** *command to build the system assembler, then issue the* **itask build cms** *command to build the CMS nucleus.*

```
295 replaces A (191)
DMSUPD181E No update files were found
ASMBLING DMSNGP
```

Note: If an error occurs during the assembly, examine the flagged statement(s), correct the statement(s) using the System Product Editor (XEDIT), and issue the **itask build cms** *command.*

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSNGP TEXT A1 CREATED
```

Nucleus Generation Prompts and Responses

Note: Each of the following prompts appears ONLY if the corresponding statement in DMSNGP is missing or empty (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?).

```
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 — This response identifies the *langid* of your system national language. The default *langid* is AMENG (American English).

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER — The default is 0 (NO); American English is not a DBCS (Double-Byte Character Set) language.

```
DMSINQ295R Language level id =
```

ENTER — The system national language does not use a level id.

DMSINQ296R Should the installation segment be used? Enter 1 (YES) or 0 (NO).

The installation segment is an optional shared discontiguous saved segment (DCSS), into which you can place frequently-used EXECs and System Product Editor (XEDIT) macros. You install the segment **after** you install your base system, but you must indicate now whether or not you are going to use it. For additional information, see "Installing CMSINST" on page 6-16.

ENTER

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 installation 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 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 disk that you specify in your response to the next prompt.

DMSINI608R IPL device address =

ENTER

The default is the address of the system disk.

DMSINI609R Nucleus (CYL or BLK) address =

nnnnn

nnnnn is the location on the 190 system disk where the new CMS nucleus is written. Enter the correct cylinder/block address for your DASD type:

Device	CYL/BLK
3330	071
3350	033
3370	29184
3375	050
3380	031

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 disk 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, refer to the description of the IPL command in the *VM/SP CP 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** *n 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

DMSEXG327I The installation DCSS could not be loaded
```

The installation 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) that you do not want to use the segment.

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

This is the default version identification. If you defined your own version identification, it appears here and each time you IPL CMS.

ENTER

Your CMS system is saved. From now on, instead of IPLing the MAINT 190 system disk, you can 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 disk, first reroute the file to MAINT's virtual reader. Issue the following commands:

```
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 disk and then receive the file from the reader. Issue:

```
access 193 r
receive spoolid cmsnuc map r
```

Once you have the load map saved on disk, you can print a copy of it by issuing:

```
print cmsnuc map r
```

- To continue the installation, go to "Step 12. Load IPCS and Complete the CUSTOMER PROFILE" on page 3-91.

Procedure, Second Level

Note: Do this step while logged on to the MAINT userid.

- 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. SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

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

- Invoke ITASK EXEC with the LOAD CMS operands:

itask load cms

```
DMSWSL409I Loading CMS SYSTEM to MAINT 190
DMSWSL409I Loading CMS BASE to MAINT 193

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 may happen if you modified SPLOAD PROFILE and changed the target disk for the CMS SYSTEM tape file from MAINT 190 to some other disk. To correct this situation, access the disk where the CMS SYSTEM files are located, issue the* **asmgend** *command to build the system assembler, then issue the* **itask build cms** *command to build the CMS nucleus.*

```
295 replaces A (191)
DMSUPD181E No update files were found
ASMBLING DMSNGP
```

Note: If an error occurs during the assembly, examine the flagged statement(s), correct the statement(s) using the System Product Editor (XEDIT), and issue the **itask build cms** *command.*

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSNGP TEXT A1 CREATED
```

Nucleus Generation Prompts and Responses

Note: Each of the following prompts appears ONLY if the corresponding statement in DMSNGP is missing or empty (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?).

```
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 — This response identifies the *langid* of your system national language. The default *langid* is AMENG (American English).

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER — The default is 0 (NO); American English is not a DBCS (Double-Byte Character Set) language.

```
DMSINQ295R Language level id =
```

ENTER — The system national language does not use a level id.

```
DMSINQ296R Should the installation segment be used?  Enter 1 (YES) or 0 (NO).
```

	The installation segment is an optional shared discontiguous saved segment (DCSS), into which you can place frequently-used EXECs and System Product Editor (XEDIT) macros. You install the segment **after** you install your base system, but you must indicate now whether or not you are going to use it. For additional information, see "Installing CMSINST" on page 6-16.
ENTER	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 installation 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 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 disk that you specify in your response to the next prompt.
DMSINI608R IPL device address =	
ENTER	The default is the address of the system disk.
DMSINI609R Nucleus (CYL or BLK) address =	
nnnnn	***nnnnn*** is the location on the 190 system disk where the new CMS nucleus is written. Enter the correct cylinder/block address for your DASD type:

Device	CYL/BLK
3330	071
3350	033
3370	29184
3375	050
3380	031

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 disk 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, refer to the description of the IPL command in the *VM/SP CP 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** *n 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

DMSEXG327I The installation DCSS could not be loaded

The installation 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) that you do not want to use the segment.

VM/SP Release *n mm/dd/yy hh:mm:ss*

This is the default version identification. If you entered your own version identification, it appears here and each time you IPL CMS.

ENTER

Your CMS system is saved. From now on, instead of IPLing the MAINT 190 system disk, you can 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 disk, first reroute the file to MAINT's virtual reader. Issue the following commands:

```
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 disk and then receive the file from the reader. Issue:

```
access 193 r
receive spoolid cmsnuc map r
```

Once you have the load map saved on disk, you can print a copy of it by issuing:

```
print cmsnuc map r
```

Step 11, Second Level

- To continue the installation, go to "Step 12. Load IPCS and Complete the CUSTOMER PROFILE" on page 3-91.

Step 12. Load IPCS and Complete the CUSTOMER PROFILE

Overview

In this step, you invoke ITASK EXEC with the LOAD IPCS operands. ITASK invokes SPLOAD EXEC to load the IPCS OBJECT tape file from the product tape to the MAINT 193 minidisk.

One of the IPCS files is named CUSTOMER PROFILE. This file contains all of the standard fields that you need to submit APARs (Authorized Program Analysis Reports). 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 data with the correct information for your installation.

If you need additional information to complete the CUSTOMER PROFILE, contact your IBM Branch Office.

For information about using IPCS, refer to the *VM Diagnosis Guide*.

Step 12

Procedure

- Invoke ITASK EXEC with the LOAD IPCS operands:

itask load ipcs

```
DMSWSL409I Loading IPCS OBJECT to MAINT 193
Ready;
```

- Use the System Product Editor (XEDIT) to edit the CUSTOMER PROFILE.

access 193 r

This command gives you read/write access to the disk where CUSTOMER PROFILE is located.

xedit customer profile

Note: The following is a sample CUSTOMER PROFILE file.

```
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. PS Rep = PSR 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. = 012-345-6789   Tie Line = 8-123-7890
------------------------------------------------------------------------
```

- Save the edited file:

file

- Go to "Step 13. Format the Remaining Base CP Minidisks" on page 3-93.

Step 13. Format the Remaining Base CP Minidisks

Overview

During the installation procedure, the following minidisks in the base CP directory do not have code loaded to them and are not associated with any particular component:

- AUTOLOG1 191
- CMSBATCH 195
- EREP 191
- OPERATNS 191
- OPERATNS 193
- MAINT 201
- MAINT 319
- CMSUSER 191
- MAINT 293
- MAINT 294
- MAINT 494
- MAINT 496
- MAINT 596

You can not use these minidisks for CMS files until you initialize them and format them into fixed-size blocks. You may invoke ITASK EXEC to assist you.

ITASK issues the CMS FORMAT command to format these minidisks in the order shown above.

Step 13

Procedure

Note: To complete this step, you must have already modified the system directory (in Step 6) to provide non-restricted passwords for the userids that "own" these minidisks. You can not format any minidisks for a userid that has a NOLOG password in the directory.

- Invoke ITASK EXEC with the BASEIDS operand:

```
itask baseids
.
.
.
Ready;
```

- Go to "Step 14. Load and Save HELP (Optional)" on page 3-95.

Step 14. Load and Save HELP (Optional)

Overview

If you do not want to load the supplied HELP files, go to "Step 15. Load, Build, and Save GCS (Optional)" on page 3-98.

To load and save HELP, you invoke ITASK EXEC with the LOAD HELP operands.

ITASK:

- Invokes 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 (from mixed case, as loaded) to uppercase. You need to convert the files if your terminals and printers do not support lowercase characters.
- Issues the SAVEFD command to initialize and save the HELP file directory information in a discontiguous saved segment (DCSS) named HELP to improve system performance. For more information about the function of the SAVEFD command and saving minidisk file directory information in a DCSS, refer to *VM/SP CMS for System Programming*.

Step 14

Procedure

- Define your virtual storage less than the address at which the HELP segment is to be loaded:

define storage 14m

The sample DMKSNT defines 14400K as the virtual storage location of the HELP segment.

```
STORAGE=14336K
CP entered;  disabled wait PSW '00020000 00000000'
```

ipl cms

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

ENTER

.
.
.

- Invoke ITASK EXEC with the LOAD HELP operands:

itask load help

Notes:

1. *If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before proceeding with the load.*

2. *Since there are many HELP files, the load operation may take 15 minutes or more.*

```
DMSWSL409I Loading HELP FILES to MAINT 19D
```

```
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 your terminals and printers do not support lowercase characters.

```
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 *n mm/dd/yy hh:mm:ss* Or your own version identification.

ENTER

- Go to "Step 15. Load, Build, and Save GCS (Optional)" on page 3-98.

Step 15. Load, Build, and Save GCS (Optional)

Overview

You must install the GCS (Group Control System) component if you plan to install SNA products or RSCS Version 2.

If you do not want to install GCS, go to "Step 16. Load TSAF (Optional)" on page 3-113.

To load GCS, you invoke ITASK EXEC with the LOAD GCS operands.

ITASK invokes SPLOAD EXEC to load the following tape files from the product tape:

- GCS INTERFACE to MAINT 193.

  This tape file contains files that enable GCS to interface with IPCS.

- GCS OBJECT to MAINT 595.

  This tape file contains the GCS object code, plus the GROUP EXEC and GCSLOAD EXEC files.

When the code is loaded, you invoke GROUP EXEC and complete a series of panels to create a GCS configuration file.

Note: If you do not have a full screen display device, you can not use GROUP EXEC, because you can not 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**.

After you create the configuration file, you invoke ITASK EXEC with the BUILD GCS operands.

ITASK:

- Modifies a copy of the GCS loadlist (GCSLOAD EXEC) and changes the entry that contains the filename of the GCS configuration file (the default is GCS) to match the filename of the configuration file that you just created (the default is also GCS), then files the modified loadlist on MAINT 295.

- Renames the filetype of the configuration file from GROUP to ASSEMBLE.

- Invokes the SPGEN ASSEMBLE procedure to assemble the configuration file.

- Invokes the SPGEN NUCLEUS procedure 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 install more than one GCS nucleus, you must re-access the GCS minidisk structure and re-invoke GROUP EXEC to create another GCS configuration file. Each configuration file must have a unique name and have an entry in the system name table (DMKSNT ASSEMBLE).

Step 15

Procedure

- Invoke ITASK EXEC with the LOAD GCS operands:

itask load gcs

Note: If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before continuing with the load.

```
DMSWSL409I Loading GCS INTERFACE to MAINT 193
DMSWSL409I Loading GCS OBJECT to MAINT 595

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 3-109.*

2. *If you do not have a full screen display device, you can not use GROUP EXEC to build the configuration file. Therefore, DO NOT set up the GROUP EXEC messages in storage. Refer to the* **VM/SP Group Control System Command and Macro Reference** *and use the build macros described there to build the configuration file manually. Then continue this step with the nucleus build operation on page 3-109.*

copyfile csimes txtameng b csiume = a

This command creates a temporary GCS message file (containing the messages for GROUP EXEC) on your A-disk that has the filename (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 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 2-5.

group *systemname*

This command assigns ***systemname*** as the filename of the GCS configuration file that you are creating and invokes the Primary Option Menu. ***systemname*** is optional at this time; if specified, it must match the SYSNAME entry for this GCS system 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.

KEY	FUNCTION
PF1 HELP	Shows information pertaining to 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've entered information, but haven't saved 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 you've 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 you've entered on the panel also exists somewhere else (in a directory entry, for instance). If you press PF9, 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, 10, 11	Not Used
ENTER	Saves information you've 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.

Figure 3-4. Function Keys Used with the GROUP EXEC Panels

```
 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

====>
```

Figure 3-5. GROUP EXEC Primary Option Menu

- To complete the Primary Option Menu:

1. **Fill in or change the SYSTEM NAME.**
   If you invoked GROUP with the ***systemname*** operand, this panel appears with the SYSTEM NAME already filled in. Make sure that the SYSTEM NAME you provide matches the SYSNAME entry for this GCS system in the system name table (DMKSNT).

2. **Verify your input before selecting another screen.**
   Press PF9 to verify that the SYSTEM NAME you entered has a valid entry in your system name table. You receive two messages: (1) tells whether the SYSTEM NAME exists and (2) says: **Enter selection number 1, 2, or 3.**

   Note: If a configuration file for this SYSTEM NAME already exists (for example, MYGCS GROUP), GROUP EXEC makes a backup copy of it, with a different filetype, before you start making changes:

   MYGCS OLDGROUP (original file, new filetype)
   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.

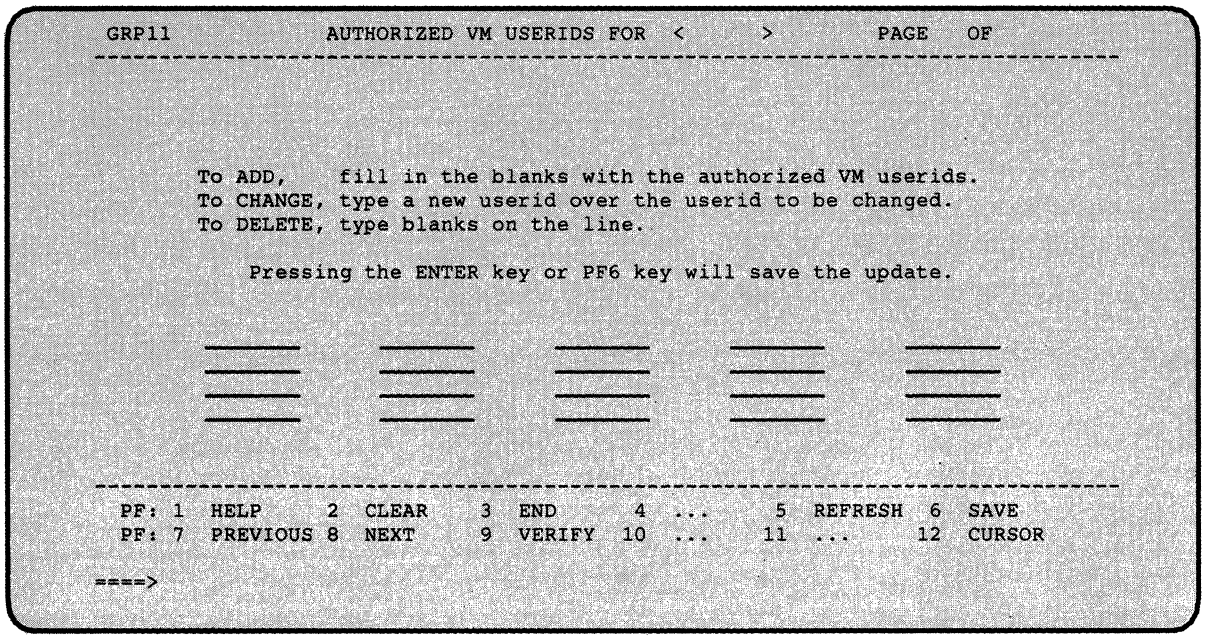


Figure 3-6. GROUP EXEC Authorized Userids Panel

- To complete the Authorized Userids panel:

1. **Type in the userid(s).**
   Press ENTER or PF6 after typing each userid to save your information and advance to the next space at the same time. Every time you press ENTER or PF6, GROUP EXEC tells you how many userids it has processed. Use PF7 and PF8 if you have more than one screenful of userids.

2. **Verify your input.**
   Press PF9. GROUP EXEC tells you how many userids on the page being viewed are valid or invalid and places arrows beside any userids not found in your VM/SP 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 userids the EXEC has 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

====>
```

Figure 3-7. GROUP EXEC Saved System Information Panel 1

- To 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; 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 userid. GROUP EXEC tells you whether that userid is valid, but it does not prevent you from saving an invalid entry.

3. **Specify a trace table size.**
   By default, 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 userids have valid entries in your VM/SP directory. If your dump receiver also appears on the other panel of Authorized VM userids, GROUP EXEC verifies it as a "Valid Authorized" userid. Make sure any new SYSTEM DISK or TRACE TABLE information is correct; 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

====>
```

Figure 3-8. GROUP EXEC Saved System Information Panel 2

- To complete the second Saved System Information panel:

  1. **Specify a MAXIMUM NUMBER.**
     Simply type a value of "1" or more, and press ENTER. 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. 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   OF
------------------------------------------------------------------------------


       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

====>
```

Figure 3-9. GROUP EXEC Automatic Saved Segment Links Panel

- To complete the Automatic Saved Segment Links panel:

  Note: Do not include CMSVSAM and CMSBAM segments here. These VSAM segments are described in 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 ENTER or PF6, the EXEC tells you how many segment names it has processed. Use PF7 and PF8 if you have more than one screen of names.

2. **Verify your input.**
   Press PF9. 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.

- Once you have provided the necessary input on all panels and returned to the Primary Option Menu for the last time, press PF3 to exit from GROUP EXEC.

- Remove the temporary GCS message file from user storage and erase it from your A-disk:

set language ameng (delete csi user

erase csiume txtameng a

- Invoke ITASK EXEC with the BUILD GCS operands:

itask build gcs *systemname*

systemname is the filename 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
DMSGLB002W File DMKSP MACLIB not found
ASMBLING systemname
.
.
.
```

Note: If you receive an assembly error, you may have to reaccess the following minidisks: **access 295 a** *(contains tailorable files),* **access 595 b** *(contains GROUP EXEC),* **access 191 c** *(contains SPGEN PROFILE),* **access 193 d** *(contains SPGEN EXEC). Then go back and check your configuration file (which ITASK has renamed to* ***systemname*** *ASSEMBLE). For information about the required fields, see "Planning for the Group Control System (GCS)" on page 2-5. Correct the file or go through the GROUP EXEC panels again to recreate the file. (If you recreate the configuration file, you must use the same* ***systemname****. After you recreate the configuration file, you must rename or erase the old* ***systemname*** *ASSEMBLE file and then rename the filetype of the new configuration file from GROUP to ASSEMBLE.) Issue the* **spgen assemble *systemname* gcs** *command to assemble the configuration file, then issue the* **spgen nucleus gcs** *command to build and save the nucleus.*

```
.
.
.
MSG FROM MAINT  :  CSIINI134I systemname HAS nnnnnn BYTES OF AVAILABLE COMMON FREE STORAGE

SYSTEM SAVED
```

Step 15

```
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 n mm/dd/yy hh:mm:ss
```

Or your own version identification.

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 on disk, issue the following commands:

access 595 e
receive *spoolid* **gcsnuc map e**

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

print gcsnuc map e

Installing Multiple GCS Systems

You can install more than one GCS system, if you defined the additional systems in DMKSNT and the system directory. To install each additional system:

1. Access the following minidisks (which were released after the previous successful build):

   access 295 a (contains tailorable files)
   access 595 b (contains GROUP EXEC)
   access 191 c (contains SPGEN PROFILE)
   access 193 d (contains SPGEN EXEC)

2. (Full screen display devices only) Set up the GROUP EXEC messages in storage:

   copyfile csimes txtameng b csiume = a
   set language ameng (add csi user

3. Create a configuration file for this nucleus:

   - If you are using a full screen display device, issue

     group *systemname*

     using a unique filename, and complete the panels to define the system parameters (see page 3-102).

   - If you are NOT using a full screen display device, refer to the *VM/SP Group Control System Command and Macro Reference.* and use the build macros to build the configuration file manually.

4. (Full screen display devices only) Remove the temporary GCS message file from user storage and erase it from your A-disk:

   set language ameng (delete csi user
   erase csiume txtameng a

5. Edit the GCS loadlist

   xedit gcsload exec

   and change the configuration file entry (`&1 &2 &3` *fn*, where *fn* is the filename of the configuration file for the previously generated GCS nucleus) to specify the filename of the configuration file for this nucleus.

Installing Multiple GCS Systems (Continued)

6. Rename the filetype of the configuration file:

   rename *systemname* group a = assemble =

7. Assemble the configuration file:

   spgen assemble *systemname* gcs

8. Build and save the new GCS nucleus:

   spgen nucleus gcs

9. Save or print the GCS load map, as indicated on page 3-110. Remember to give each load map file a different name.

- When you complete your GCS installation, go to "Step 16. Load TSAF (Optional)" on page 3-113.

Step 16. Load TSAF (Optional)

Overview

If you do not want to load the TSAF (Transparent Services Access Facility) component, go to "Step 17. Load Source Code (Optional)" on page 3-114.

To load TSAF, you invoke ITASK EXEC with the LOAD TSAF operands.

ITASK invokes SPLOAD EXEC to load the following tape files from the product tape:

- TSAF INTERFACE to MAINT 193.

  These files enable TSAF to interface with IPCS.

- TSAF OBJECT to MAINT 492.

  This tape file contains the ready-to-run TSAF module. No build is required.

For more information about TSAF, refer to the *VM/SP Transparent Services Access Facility Reference.*

Procedure

■ Invoke ITASK EXEC with the LOAD TSAF operands:

itask load tsaf

Note: If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before continuing with the load.

```
DMSWSL409I Loading TSAF INTERFACE to MAINT 193
DMSWSL409I Loading TSAF OBJECT to MAINT 492

Ready;
```

■ Go to "Step 17. Load Source Code (Optional)" on page 3-114.

Step 17. Load Source Code (Optional)

Overview

Source code for the CP, CMS, and IPCS components is provided on the product tape. Source code for the GCS and TSAF components and selected CMS modules is provided on a separately-orderable feature tape.

If you do not want to load source, go to "Step 18. Update the SYSTEM NETID File (Optional)" on page 3-117.

In this step, you invoke ITASK EXEC with specific operands to identify the source tape files that you want to load.

ITASK invokes SPLOAD EXEC to load the specified source tape files to the minidisks defined in SPLOAD PROFILE. The default minidisks are:

- MAINT 394 for CP SOURCE
- MAINT 393 for CMS SOURCE and CMSFT SOURCE
- MAINT 497 for IPCS SOURCE
- MAINT 597 for GCS SOURCE
- MAINT 493 for TSAF SOURCE.

Note: The MAINT 597 and 493 minidisks are not defined in the sample directory that you loaded from the product tape in Step 5. If you plan to load GCS or TSAF source code, and you did not define the required minidisk(s) when you tailored the directory in Step 6, you must define the minidisk(s) in this step before you invoke ITASK.

All source files are loaded down to minidisk 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, refer to the *VM/SP CMS Command Reference*. The MAINT 394, 393, and 497 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 your files, you may need to redefine the size of these minidisks.

Procedure

- Invoke ITASK EXEC with the operands that identify the source tape file that you want to load from the product tape:

itask load *group*src

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) used in SPLOAD PROFILE. Enter: **cp**, **cms**, or **ipcs**.

Note: If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before continuing with the load.

DMSWSL409I Loading *group* **SOURCE to MAINT** *vdev*

- Issue 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.
- Invoke ITASK EXEC with the operands that identify the source tape file that you want to load from the source feature tape:

Note: Before you invoke ITASK to load GCS or TSAF source code, make sure that the target minidisk is defined in the directory. If you did not define the MAINT 597 minidisk (for GCS) or the MAINT 493 minidisk (for TSAF) when you tailored the directory in Step 6, you must edit the directory now to define one or both of these minidisks (see page 3-48). After you define the minidisk(s), issue the DIRECT command to place the updated directory on line, then log off and log on to make the changes effective.

itask load *group*src

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) used in SPLOAD PROFILE. Enter: **gcs**, **tsaf**, or **cmsft**.

Note: If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before continuing with the load.

Step 17

DMSWSL409I Loading *group* **SOURCE to MAINT** *vdev*

- Issue 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 18. Update the SYSTEM NETID File (Optional)" on page 3-117.

Step 18. Update the SYSTEM NETID File (Optional)

Overview

If you created an RSCS (Remote Spooling Communications Subsystem) networking virtual machine in the system 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 19. Install EREP (Optional)" on page 3-120.

The records in the SYSTEM NETID file have the format

cpuid nodeid netid

where:

- *cpuid* is the CPU serial number.
- *nodeid* is the local nodeid 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 userid of the RSCS virtual machine, as defined in the system directory.

When you issue 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 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.

Step 18

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.

- Use the System Product Editor (XEDIT) to 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 parm savesys cms
```

```
SYSTEM SAVED
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

- Go to "Step 19. Install EREP (Optional)" on page 3-120.

Step 19. Install EREP (Optional)

Overview

If you ordered EREP (the Environmental Recording Editing and Printing program), this licensed program is supplied on a separate distribution tape.

If you did not order EREP, go to "Step 20. Do a Full System Backup (Recommended)" on page 3-122.

In this step, you invoke INSTFPP EXEC to load the EREP files to the EREP minidisk (MAINT 201 is the default).

Procedure

- Mount the EREP tape.
- Invoke INSTFPP EXEC to load the EREP code:

instfpp 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 cuu (***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 20. Do a Full System Backup (Recommended)" on page 3-122.

Step 20. 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 DDR (DASD Dump/Restore) program. For more information about DDR, refer to *VM/SP CP for System Programming*.

To obtain maximum protection if you plan to apply preventive or corrective service, you may 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 userid.

- Mount and ready a backup tape.
- Invoke DDR to dump the DASD volume to tape. Follow this sequence for each volume that you want to dump:

ddr

ENTER:

sysprint cons

This command directs system output to the console.

ENTER:

input *vdev type label*

vdev is the virtual address of the DASD volume.

type is the device type of the DASD.

label is the label of the DASD volume.

ENTER:

output *vdev type* (mode *bpi* compact

vdev is the virtual address of the tape drive.

type 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, refer to *VM/SP CP for System Programming*.

ENTER:

dump cpvol

This command dumps cylinder 0 (block 0-15 in an FBA device) and all disk space allocated as PERM, DRCT, and OVRD onto the tape.

DUMPING *label*

.
.
.

END OF DUMP

Step 20

ENTER

```
END OF JOB
```

■ Go to "Related Procedures" on page 3-125.

Related Procedures

You have now completed your base VM/SP installation. However, there are several related procedures that you may also want to complete:

- **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 discontiguous saved segments (DCSS).**

  For procedures to install the CMSDOS, CMSBAM, CMSVSAM, CMSAMS, and CMSINST segments, see Chapter 6, "Installing Discontiguous 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, refer to *VM System Facilities for Programming.*

- **Install optional feature program products.**

  If you ordered optional feature program products, see Chapter 8, "Installing Optional Feature Program Products on VM/SP" on page 8-1.

- **Apply Service.**

  If a PUT (program update tape) was included with your shipment, you may want to apply some or all of the service updates that it contains. For information about applying preventive and/or corrective service, see "Part 3. VM/SP Service."

Chapter 4. Installing VM/SP Using an Existing VM System

The RECOMMENDED procedure to install VM/SP on an existing (prior release) VM system is the Starter System procedure described in Chapter 3. The Starter System procedure requires fewer entries and decisions. However, if you do not have enough DASD available to use the Starter System, or if your system configuration is substantially different from the configuration defined by the sample files, you can use the procedure described in this chapter.

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 userid to do the installation. Therefore, all new or modified installation and system generation minidisks in the system directory are defined for MAINT.
- You understand the page layout and text formats used in this book (see "The Format Used in This Book" on page 1-12).

Migration Considerations

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.
- VM/SP requires at least 768K real storage.

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:

- Migrating spool files
- Identifying your DASD requirements
- 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 on your system.

Important additions and changes to the installation procedure from previous releases include:

- **New installation and system generation tools and profiles**

  See "Tools and Profiles for Installation and System Generation" on page 1-9, and Appendix C, "Sample Installation/System Generation Profiles" on page C-1.

- **New and modified minidisks defined for MAINT**

  See Appendix A, "Minidisks Reserved for the MAINT Userid" on page A-1.

- **New procedure to migrate spool files**

  See Appendix E, "Migrating Spool Files Using SHUTDOWN/WARM IPL" on page E-1.

Preparation

Before you begin the installation procedure:

- 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 may want to use the COMPACT option on DDR to save time and tape. Refer to **VM/SP CP for System Programming** *for assistance.*

- Have a printout of your current system directory readily available.

- If you have spool files that you want to migrate:
  - You can use the SPTAPE command to save the spool files on tape. For more information about using SPTAPE, refer to the *VM/SP CP Command Reference.*
  - If you are migrating from VM/SP Release 3 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.

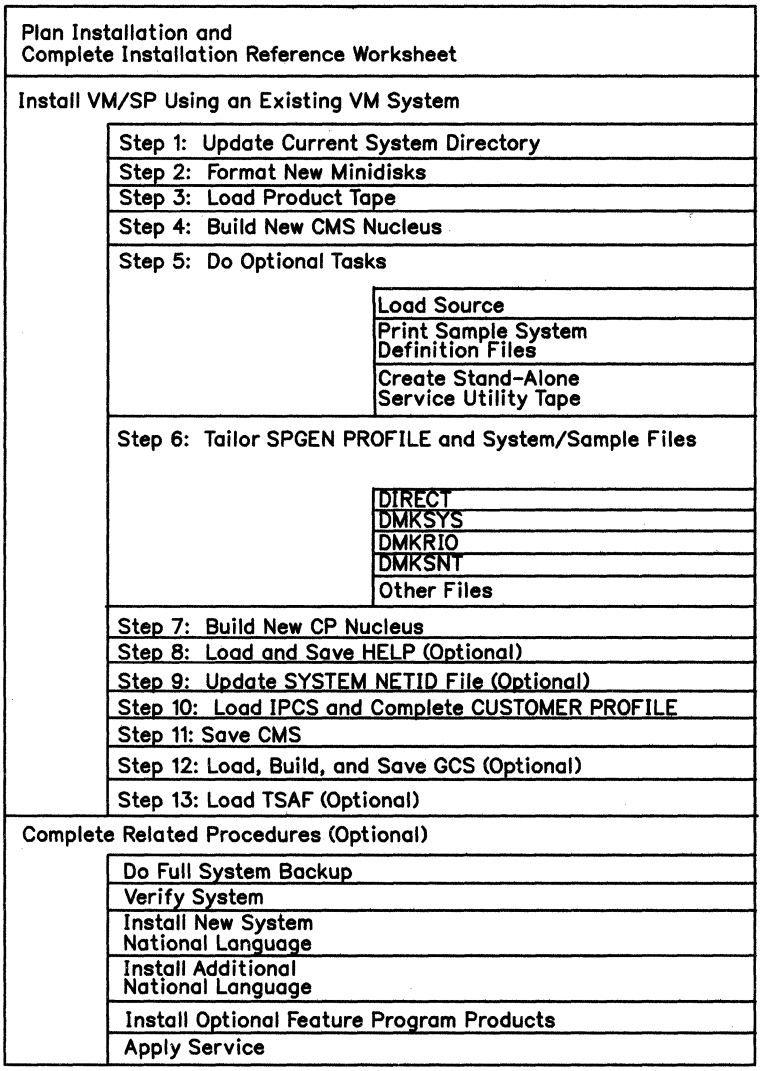


Figure 4-1. Installing VM/SP Using an Existing VM System

Step 1. Modify the Current System Directory

Overview

- Edit the current system directory
- Define a new system minidisk (MAINT 490)
- Define other minidisks (as required)
- Adjust minidisk sizes (as required)
- Make sure that each user has a link to the HELP disk
- Place the updated directory on line
- Log off, then log back on as MAINT

Procedure

■ Edit the current system directory:

xedit ***fn ft***

■ Define MAINT's virtual storage as 16M.

■ Define a minidisk to replace the current MAINT 190 system disk. The examples in this procedure use MAINT 490.

Note: Defining the 490 minidisk on the same physical volume as the 190 minidisk keeps the installation procedure simpler.

■ Define other minidisks as required for the components and products that you plan to install, such as the Group Control System (GCS), the Transparent Services Access Facility (TSAF), the Environmental Recording Editing and Printing program (EREP), etc. See Figure A-1 on page A-1.

■ Compare your current minidisk sizes with the requirements for this release and make any necessary adjustments. See Figure A-2 on page A-3.

■ (Current VM/SP Release 1 users only) Make sure that each user has a link to the default system HELP files disk, 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.

Step 1

- File the updated directory and place it on line:

file

direct *fn ft*

- To make these changes effective, log off, then log back on as MAINT.

Step 2. Format the New Minidisks

Overview

- Determine the block size of the MAINT 190 minidisk
- Format the new MAINT 490 minidisk
- Format MAINT 490 with the RECOMP option
- Format any other new minidisks that you defined in Step 1
- 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 *n*k

fm is any available filemode.

n is the block size of the 190 disk.

■ Format MAINT 490 with the RECOMP option:

format 490 *fm nnnnn* (recomp blksize *n*k

fm is the same filemode that you used in the previous command.

nnnnn is the starting cylinder/block on 490 where the new CMS nucleus will reside, according to the device type and minidisk size listed in Figure 4-2.

n is the block size of the 190 disk.

Device Type	190 Allocation	Nucleus Size	nnnnn (CMS)
3330	83 cyl	12 cyl	71 cyl
3340	205 cyl	26 cyl	179 cyl

Figure 4-2 (Part 1 of 2). CMS Nucleus Size and Location

Device Type	190 Allocation	Nucleus Size	nnnnn (CMS)
3350	39 cyl	6 cyl	33 cyl
3375	58 cyl	8 cyl	50 cyl
3380	37 cyl	6 cyl	31 cyl
FBA	34312 blk	5128 blk	29184 blk

Figure 4-2 (Part 2 of 2). CMS Nucleus Size and Location

Note: If you try to use the current CMS 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 minidisk (490) is recommended.

- Format any other new minidisks that you defined in Step 1:

format *vdev fm*

vdev is the virtual address of the minidisk.

fm is any available filemode.

- 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

- Mount the product tape on device 181
- Load the installation tools and profiles to MAINT 191
- Load the system generation tools to MAINT 193
- Load the sample files to MAINT 295
- Load the CP object code and the sample files for the printer universal character set and font offset buffer to MAINT 194
- Load the CMS system code to MAINT 490
- Load the CMS base code to MAINT 193

Procedure

Note: See page 1-7 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 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 contains:

ITASK EXEC
SPLOAD EXEC
SPLOAD PROFILE
PROFILE EXEC (for MAINT)
SPGEN PROFILE
INSTFPP EXEC.

- Load the system generation tools to MAINT 193:

access 193 b

vmfplc2 load * * b

This command loads the next tape file, SYSGEN TOOLS, which includes:

SPGEN EXEC
UTILITY EXEC
SNTMAP EXEC
DISKMAP EXEC
DIRECGEN EXEC
$MSG4I$ EXEC
$DASD$ CONSTS.

- Load the sample files to MAINT 295:

access 295 c

vmfplc2 load * * c

This command loads the next tape file, SYSTEM SAMPLES, which contains:

DMSNGP SAMPLE
DMKBOX ASSEMBLE
DMKFCB ASSEMBLE
RPWLIST DATA.

Warning: If your current CP includes the ADRP (Auto-Deactivation of Restricted Passwords) feature, the RPWLIST DATA file that you just loaded may be different from your previous RPWLIST DATA file (see Appendix B, "Restricted Logon Passwords" on page B-1). Issuing the DIRECT command causes the system to search the directory for logon passwords that match the list contained in the first available RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory before the directory is placed on line. If your current CP does not include ADRP, the password checking begins after you build the new CP (which includes ADRP) in Step 7.

vmfplc2 load * *devtype[-mm]* c

devtype[-mm] is the device type and model (if any) of your system residence DASD. This command loads the DASDMODL SAMPLES tape file for your DASD type and model. This tape file contains:

DIRECT *devtype[-mm]*
DMKSNT *devtype[-mm]*
EXTENTS *devtype[-mm].*

vmfplc2 load * *devtype* c

devtype is the device type of your system residence DASD. This command loads the DASDTYPE SAMPLES tape file for your DASD type. This tape file contains:

DMKRIO *devtype*
DMKSYS *devtype.*

- Load the CP object code and the sample files for the printer universal character set and font offset buffer to MAINT 194:

access 194 d

vmfplc2 load * * d

This commmand loads the next tape file, CP OBJECT, which contains the CP object code and the following sample files:

DMKPIA ASSEMBLE
DMKPIB ASSEMBLE
DMKUCB ASSEMBLE
DMKUCC ASSEMBLE
DMKUCS ASSEMBLE.

- Load the CMS system code to MAINT 490:

access 490 t

You must access 490 with a filemode below S. 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, CMS SYSTEM, which contains EXECs and MODULEs used by the general CMS user.

- Load the CMS base code to MAINT 193:

access 193 u	You must access 193 with a filemode below S. 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, CMS BASE, which contains TEXT files and other items used primarily by system programmers.

Tailoring SPLOAD PROFILE

Other tape files (such as source, HELP, IPCS, and GCS) are loaded later in this procedure using the SPLOAD EXEC installation tool, which refers to the SPLOAD PROFILE for load instructions. If you want to change the default userid and minidisk addresses, use the System Product Editor (XEDIT) to tailor SPLOAD PROFILE. See "Sample SPLOAD PROFILE" on page C-1.

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.

If you change SPLOAD PROFILE, you may not be able to use the ITASK EXEC installation tool, because ITASK expects to find certain information on specific minidisks. There are no safeguards in the code to check the validity of your changes.

Step 4. Build a New CMS Nucleus

Overview

- Rename the sample DMSNGP file
- Modify DMSNGP
- Assemble DMSNGP
- Copy the DMSNGP TEXT file to MAINT 490
- Build the new CMS nucleus
- IPL your reader to write the new CMS nucleus on disk
- Answer the nucleus generation prompts (if displayed)
- IPL the new CMS nucleus
- Save and print the CMS load map (optional)
- Rebuild the system assembler using modules for the new release

Procedure

Warning: To prevent updates for the prior release from being applied to this release, make sure that NO service minidisks are accessed.

Note: Due to expansion of the CMS nucleus, you may need to increase the size of your virtual machine.

■ Rename and edit the sample CMS nucleus generation profile (DMSNGP):

```
rename dmsngp sample c = assemble =

xedit dmsngp assemble c
```

DMSNGP defines responses to the system prompts that are generated when you build and save 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 generate the nucleus. However, to ensure that all necessary system values are defined, if a DMSNGP statement is missing or empty (and that parameter has no default value in the DEFNUC macro), or if a DMSNGP statement contains a question mark (?), the corresponding prompt does appear when you generate the nucleus.

DMSNGP is supplied with most responses predefined. For more information about DMSNGP and the DEFNUC macro, refer to the *VM/SP Planning Guide and Reference.*

■ Make the following changes to DMSNGP:

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 **?** or **NO**.

   Note: If you change the statement to ?, you will receive a prompt during the nucleus build asking if you want to save the system. You must enter **0** *(NO) at that time.*

   You DO NOT want to save CMS after you create the nucleus, because you have not yet updated the CMS entry in the system name table (DMKSNT).

2. Change the IPLADDR statement to **?** or **490**.

   Note: If you change the statement to ?, you will receive a prompt during the nucleus build asking for the IPL device address. You must enter **490** *at that time.*

   You want to write the new CMS nucleus to the MAINT 490 minidisk that you defined and formatted in previous steps.

3. Complete the CYLADDR statement to indicate the cylinder/block address on 490 where the nucleus is written, according to your DASD type:

Device	CYL/BLK Address
3330	071
3340	179
3350	033
3375	050
3380	031
FBA	29184

4. Change other statements, such as the version identification and the installation heading, to reflect the responses that you want to use. You must place single quotes around any character string that you enter as the new VERSION or INSTID.

- File the modified DMSNGP file and assemble it:

```
file
release t
access 490 b/a * maclib
```

To assemble DMSNGP, you need to have access to the DEFNUC macro located on MAINT 490.

```
vmfasm dmsngp dmssp
.
.
.
```

```
DMSNGP TEXT A1 CREATED
```

- Copy the TEXT file to the new MAINT 490 minidisk:

```
access 490 t
copyfile dmsngp text a = = t (replace
```

- Issue the VMFLOAD command to build the new CMS nucleus:

```
purge rdr class i
spool pun * class i nocont
spool prt *
vmfload cmsload dmssp
SYSTEM LOAD DECK COMPLETE
PUN FILE nnnn TO MAINT COPY 0001 NOHOLD
```

- IPL your reader to write the new CMS nucleus on disk:

```
spool rdr class i
ipl 00c
```

Nucleus Generation Prompts and Responses

Note: Each of the following prompts appears ONLY if the corresponding statement in DMSNGP is missing or empty (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?).

```
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 — This response identifies the *langid* of your system national language. The default is American English (AMENG).

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER — The default is 0 (NO); American English is not a DBCS (Double-Byte Character Set) language.

```
DMSINQ295R Language level id =
```

ENTER — This response designates the discontiguous saved segment (DCSS) level to be loaded. The system national language does not use a DCSS.

```
DMSINQ296R Should the installation segment be used?  Enter 1 (YES) or 0 (NO).
```

The installation segment is an optional shared discontiguous saved segment (DCSS) into which you can place frequently-used EXECs and System Product Editor (XEDIT) macros. You install this segment after you install your base system. For additional information, see "Installing CMSINST" on page 6-16.

ENTER (or **1**) or **0** — The default is 1 (YES). Enter **0** if you want to omit 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 an installation segment name consisting of one to eight alphanumeric characters, 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 disk that you specify in your response to the next prompt.

DMSINI608R IPL device address =

490

490 is the new 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	CYL/BLK Address
3330	071 cyl
3340	179 cyl
3350	033 cyl
3375	050 cyl
3380	031 cyl
FBA	29184 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 disk 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, refer to the description of the IPL command in the *VM/SP CP Command Reference*.

DMSINI611R Enter version identification:

The version identification is displayed each time you IPL the CMS system you are now generating. 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** *n mm/dd/yy hh:mm:ss*.

ENTER or ***version***

DMSINI612R Enter installation heading:

You can enter up to 64 descriptive characters to serve as an installation heading at the beginning of each output file, or you can press **ENTER** to accept the default heading, **VM/SP CONVERSATIONAL MONITOR SYSTEM**.

ENTER or ***heading***

End of Nucleus Generation Prompts and Responses

VM/SP Release *n mm/dd/yy hh:mm:ss*

This is the default version identification. If you defined your own version identification, it appears here and each time you IPL 190 or IPL CMS.

ENTER

- IPL the new CMS nucleus:

define 190 590

This command redefines the old CMS to 590.

define 490 190

This command redefines the new CMS to 190.

ipl 190

You must IPL the new CMS to use the new system generation tools.

PRT FILE *spoolid* **TO MAINT COPY 0001 NOHOLD**

The *spoolid* in this message identifies the CMS load map file. Record the *spoolid* if you plan to save or print the load map.

VM/SP Release *n mm/dd/yy hh:mm:ss*

Or your own version identification.

access 190 a

access 193 b	The system generation tools are on this disk.

Saving and Printing the CMS Load Map

The CMS load map is in MAINT's virtual reader. To save the load map on disk, issue the following commands:

#cp spool prt nohold
receive *spoolid* cmsnuc map b

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

print cmsnuc map b

- Rebuild the system assembler using modules for the new release:

asmgend

```
ASSEMBLE XF GEND PROC

Enter target disk mode for assembler modules.
Defaults to S-disk if none entered.

ENTER

ASSEMBLE XF GEND COMPLETE
```

Step 5. Do Optional Tasks

Overview

- Load the source code (optional)
- Print the sample system definition files (optional)
- Create a stand-alone service utility tape (optional)

Procedure

■ Load the source code (optional).

Source code for the CP, CMS, and IPCS components is provided on the product tape. Source code for the GCS and TSAF components and selected CMS modules is provided on a separately-orderable feature tape.

Invoke SPLOAD EXEC to load the source tape files. Default minidisk destinations for these tape files are specified in SPLOAD PROFILE, as follows:

MAINT 394 for CP SOURCE
MAINT 393 for CMS SOURCE and CMSFT SOURCE
MAINT 497 for IPCS SOURCE
MAINT 597 for GCS SOURCE
MAINT 493 for TSAF SOURCE.

If you want to load a source tape file to a different address, modify the SPLOAD PROFILE entry before you issue the SPLOAD command (see "SPLOAD PROFILE Syntax" on page 15-33).

Note: Source code is supplied and loaded down to minidisk in **packed** *format. To unpack your source files, use the COPYFILE command with the UNPACK option. For more information, refer to the* **VM/SP CMS Command Reference**. *The MAINT 394, 393, and 497 minidisk sizes recommended in Figure 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 your files, you may need to redefine the size of these minidisks.*

sploadgroup 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 SPLOAD PROFILE: **cp**, **cms**, or **ipcs** on the product tape; **gcs**, **tsaf**, or **cmsft** on the source feature tape.

Issue this command for each source tape file that you want to load.

Note: If you are loading a minidisk that has not been formatted yet, SPLOAD EXEC formats the minidisk before continuing with the load.

DMSWSL409I Loading *group* **SOURCE to MAINT** *vdev*

- Print copies of the sample files (optional). Having printed copies of the sample files may be helpful when you tailor the system definition files in Step 6.

  Use your normal method for printing files from disk to print copies of the files that you want. Remember that some sample files have DASD-specific filetypes (see page 4-10).

- Create a stand-alone service utility tape (optional).

  If you want 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)

  see "UTILITY EXEC" on page 15-36 for more information.

Step 6. Tailor SPGEN PROFILE and the System/Sample Files

Overview

- Tailor SPGEN PROFILE
- Tailor the current system directory
- Tailor the current DMKSYS ASSEMBLE
- Tailor the current DMKRIO ASSEMBLE
- Tailor the **sample** DMKSNT
- Tailor the other sample files as required

Procedures

This section provides only guidance and suggestions. Use the System Product Editor (XEDIT) to tailor these files as required for your installation. For additional information, refer to the *VM/SP Planning Guide and Reference.*

SPGEN Considerations

- Tailor SPGEN PROFILE if you want to change the supplied nucleus generation information.

  For information about the contents of SPGEN PROFILE, see "SPGEN PROFILE Syntax" on page 15-26 and "SPGEN PROFILE Defaults" on page 15-29.

  Make sure that the minidisk access order that follows the SETUP keyword reflects your actual system configuration.

  See Appendix D, "Special Options for CP" on page D-1, for more information about the following CP options:

  - Small CP
  - CP FRET Trap
  - Virtual=Real

Directory Considerations

- Compare your current directory to the sample directory in the listing that you printed in Step 5.

- Update your current directory by copying any new or revised information provided in the sample directory (DIRECT *devtype[-mm]*) that you want to include in your new system.

Consider the following changes:

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

  As a system security measure, the new CP includes the ADRP (Auto-Deactivation of Restricted Passwords) feature. After you build the new CP (or if your current CP already includes ADRP), issuing the DIRECT command to process the 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. 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 that has a NOLOG password.

  Change all restricted logon passwords in your directory to non-restricted passwords. For a list of the restricted passwords in the RPWLIST DATA file, see Appendix B, "Restricted Logon Passwords" on page B-1.

- **Updating the MAINT Userid**

  Update the MAINT userid 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 Userids**

  Some of the userids defined in the sample directory, such as GCS and TSAFVM, may not be included in your current system. You may want to copy some or all of these entries to your current directory.

  If you plan to install additional program products (such as RSCS) that require system userids 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 GCS Userid**

  If you are installing the GCS component, you may want to change the specification of the sample GCS recovery machine (userid GCS).

- **Modifying the TSAFVM Userid**

  If you are installing the TSAF component, you may want to change the specification of the sample TSAF virtual machine (userid TSAFVM). Remember that all userids in all systems in the TSAF collection must be unique.

- **Restructuring User Classes**

  The user classes specified in the sample directory are IBM-standard. 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. *VM/SP CP for System Programming* provides a procedure for user class restructure.

- **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.

- **Defining the Real VMSRES Address**

  In Step 7, you can create a backup tape containing an IPLable copy of your CP nucleus. However, to make this backup tape usable, you **must** change the following directory statements to reflect the **real** address of your VMSRES volume:

  ```
  DIRECTORY 123
  MAINT MDISK 123
  ```

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

  diskmap ***fn ft***

  DISKMAP produces a file called DIRECT DISKMAP, which is placed on your A-disk. For more information about DISKMAP EXEC, see page 15-9.

Warning: DO NOT issue the DIRECT command to place the updated directory on line at this time. Since you are using your current CP to build the new CP, you must continue to use the directory associated with your current CP until the new CP is built.

DMKSYS Considerations

- Compare your current DMKSYS file to the sample file in the listing that you printed in Step 5.
- Update your DMKSYS file by copying any new or revised information provided in the sample file (DMKSYS *devtype*) that you want to include in your new system.

  Consider the following changes:

  - **Allocating Space for Paging**

    The sample DMKSYS file does not provide space for paging. If you plan to use the sample file to build your system, TEMP space will be used for paging unless you define paging space. Refer to the *VM/SP Planning Guide and Reference* for information about calculating paging space requirements and allocating space for paging in the SYSOWN macro.

  - **Restructuring User Classes**

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

  - **Defining the GCS Volume**

    If you plan to install GCS on its own DASD volume, make sure that the GCS volume label is entered in the SYSOWN 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*. Refer to the *VM/SP Planning Guide and Reference* for the proper format for this entry.

  - **Installing an AP or MP System**

    If you plan to generate an AP or MP system, you MUST change the SYSCOR macro. Refer to the *VM/SP Planning Guide and Reference* for more information about this macro.

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

- Compare your current DMKSNT file to the sample file in the listing that you printed in Step 5.

  Due to the number of changes made to DMKSNT for this release (such as the relocation of the CMS saved system, the deletion of CMSL, the addition of the CMSINST and HELP saved segments), you should use the sample DMKSNT file (DMKSNT *devtype[-mm]*) and copy over any local changes from your current file.

- Tailor the sample file to define your installation.

  Warning: Make sure that your changes do not cause overlay of other products on your DASD. Also make sure that no segment overlays another segment in virtual storage unless the segments are NEVER used at the same time. Figure 6-1 on page 6-3 shows the layout of 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 discontiguous saved segments on the volume may be necessary. For reference, you may want to load and print the sample DMKSNT file for the other DASD type. On the product tape, look for the DMKSNT *devtype[-mm]* file.

  - **Modifying the System Disk Pointers**

    You MUST 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 saved segments, you should add the DMKSNT entries now. Otherwise, you will have to reassemble DMKSNT and recreate 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.

- 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.

- Rename or erase the current file, then rename the sample file to become the current file.

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

  sntmap dmksnt

  SNTMAP produces two output files on your A-disk:

  - DASD SNTMAP lists the system/segment names and DASD allocations in order of occurrance on each DASD volume.

  - MEMORY SNTMAP lists the system/segment names and DASD allocations according to location in virtual memory.

  For more information about SNTMAP EXEC, see page 15-21.

Other Considerations

- If you want to change the forms control buffer macros, tailor DMKFCB ASSEMBLE. Refer to *VM/SP CP for System Programming* for directions.
- If you want to change the design or contents of the logon logo, tailor DMKBOX ASSEMBLE.
- If you have defined new CP commands, refer to *VM/SP CP for System Programming* 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.

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
```

Step 7. Build a New CP Nucleus

Overview

- Backup the current CP nucleus and $DIRECT$ area
- Rename or erase any duplicate system files
- Assemble the sample files
- Build the new CP nucleus
- Save and print the CP load map (optional)
- Place the tailored system directory on line
- Shut down the production system
- IPL the VMSRES volume
- Log on to the MAINT userid
- Save an IPLable CP nucleus on tape (optional)
- Define and save an additional CP nucleus (optional)

Procedure

- Use DDR to backup your current CP nucleus and $DIRECT$ area. For more information about using DDR, refer to *VM/SP CP for System Programming*.

- To avoid having duplicate files, rename or erase any previous-release system 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.** You loaded these sample files to the 295 disk; SPGEN EXEC accesses this disk as N.

- Assemble the files that you tailored in Step 6. Assemble the sample DMKBOX, DMKFCB, and DMKSNT even if you did not tailor these files. Set up the necessary minidisk access order and select ONE of the following methods to assemble the files:

  - Invoke VMFASM EXEC to assemble each file. For more information about VMFASM EXEC, see page 15-38.

  - Invoke SPGEN EXEC with the ASSEMBLE operand to assemble each file. SPGEN EXEC obtains the control file name from SPGEN PROFILE and invokes VMFASM EXEC to do the assembly. This method is shown below.

spgen assemble *fn*

fn is the filename of the file that you want to assemble. The filetype must be ASSEMBLE.

```
DMSUPD181E No update files were found
ASMBLING fn
```

Note: If any errors occur during the assembly:

1. *Examine the flagged statement(s).*
2. *Correct the statement(s) using the System Product Editor (XEDIT).*
3. *Reissue the* **spgen assemble *fn*** *command to assemble the file.*

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
fn TEXT A1 CREATED
```

- Complete the previous sequence for DMKBOX, DMKFCB, DMKSNT, and each sample file that you tailored.
- Build the new CP nucleus. Select ONE of the following methods:
  - Issue the VMFLOAD command with the proper CP loadlist and control file names. For more information about the VMFLOAD command, see page 15-53.
  - Invoke SPGEN EXEC with the NUCLEUS CP operands. SPGEN EXEC obtains the CP loadlist and control file names from SPGEN PROFILE and issues the VMFLOAD command to build the nucleus. This method is shown below.

```
spgen nucleus cp
.
.
.
NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=nnn  , LAST CYL/BLK USED=nnn
```

Note: If any errors occur during the load:

1. *Inspect the load map and the virtual PSW. A loader error may be indicated on the listing or in the PSW. Refer to* **VM/SP System Messages and Codes** *for a list of the loader wait state codes.*
2. *Correct the error, then reissue the* **spgen nucleus cp** *command.*

The **NUCLEUS LOADED** message gives the starting and ending cylinders/blocks for the nucleus. Remember that you are using the **current** CP directory to create this nucleus. Because you made changes to the current directory in Step 2, you should verify that this nucleus has not expanded into another allocated area such as TEMP, WARM START, etc., since unpredictable results may occur.

An easy way to check this is to look at VMUSERS DISKMAP (which you created after you updated the directory) and make sure that **LAST CYL/BLK USED**, as displayed above, does not exceed the $CPNUC$ ending cylinder/block as defined in the directory. To look at this file, you must first issue the **#cp ipl 190** command.

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

This is the message that you receive if the nucleus loaded correctly. However, on many processors this message does not display automatically. If the PSW message does not display within about one minute, enter **#cp display psw**.

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 disk, issue the following commands:

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

After **VM READ** appears in the corner of your screen, issue:

```
ENTER
access 194 c
query rdr
```

```
MAINT spoolid M PUN nnnnnnnn 001 NONE STANDARD
```

```
receive spoolid cpnuc map c
```

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

```
print cpnuc map c
```

#cp ipl 190

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.

access 295 b

- Place the updated system directory on line:

access 190 c

Since the new DIRECT MODULE file is on the new 190 disk, you must make sure that you have this disk accessed ahead of the old system disk (now 590).

Warning: Issuing the DIRECT command causes the system to search the directory for logon passwords that match the list of restricted passwords in the RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory. Before you issue this command, make sure that your directory contains only non-restricted passwords (see Appendix B, "Restricted Logon Passwords" on page B-1).

direct vmusers direct

- 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. **Be sure to wait until all devices are drained, as indicated by the Ready message.**

Ready;

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 n, 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**, since 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 may 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 userid.

```
DMKCPJ952I nnnnK system storage
```

For some processors, *nnnn***K** may be less than your actual system storage.

```
.
.
.
```

```
DMKCPI966I Initialization complete
```

- Log on to the MAINT userid 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

Once you have logged on as MAINT, you can save a copy of your CP nucleus on tape for recovery purposes. Invoke 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 the system directory (see page 4-24) and in DMKSYS (see page 4-25).*

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, refer to the *VM/SP Planning Guide and Reference.*

Step 8. Load and Save HELP (Optional)

Overview

- Load the system HELP files
- Convert the files to uppercase (if required)
- Save the HELP segment (optional)

Procedure

■ Load the HELP FILES tape file:

ipl 190

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

sploadhelp 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 default target minidisk 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 (see "SPLOAD PROFILE Syntax" on page 15-33).

```
DMSWSL409I Loading HELP FILES to MAINT 19D
```

Note: Since there are many HELP files, the load operation may take 15 minutes or more.

```
Ready;
```

Converting the HELP Files to Uppercase

If your terminals or printers do not support lowercase characters, you can convert the HELP files from mixed case (as loaded) to uppercase by issuing the COPYFILE command with the UPCASE option for each file.

Saving the HELP File Directory Information in a DCSS

To improve system performance, you may want to save the HELP file directory information in a discontiguous saved segment (DCSS).

The sample DMKSNT contains a NAMESYS entry for a DCSS named HELP. To use this segment to save the directory information for HELP files loaded to the default MAINT 19D minidisk:

1. Define your virtual storage **less** than the address at which the HELP segment is to be loaded. For example, the sample DMKSNT defines the load address as 14400K; therefore, define your storage as 14M (14336K).

   Because CMS has not been saved yet, to remove the disabled wait condition after you define your storage, enter:

   ipl 190

2. Issue 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 disk.

3. Define your virtual storage back to 16M, then issue:

   ipl 190

For more information about using the SAVEFD command to save minidisk file directory information in a DCSS, refer to *VM/SP CMS for System Programming*.

Step 9. Update the SYSTEM NETID File (Optional)

Overview

If you have an RSCS (Remote Spooling Communications Subsystem) networking virtual machine defined in the system directory, you should now update the SYSTEM NETID file located on the MAINT 190 minidisk. This file is referenced when you issue CMS commands to communicate via 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 nodeid 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 userid of the RSCS virtual machine, as defined in the system directory.

When you issue 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 issue the IDENTIFY command.

2. The IDENTIFY command issues the QUERY CPUID command to retrieve the CPU serial number and searches the SYSTEM NETID file for a matching serial number.

3. The IDENTIFY command issues the QUERY USERID command to retrieve the node identification and 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 = FF*ssssssdddd*0000

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

- Load the IPCS object code to MAINT 193
- Edit the CUSTOMER PROFILE file
- Replace the sample data with the correct information for your installation

Procedure

- Load the IPCS object code (IPCS OBJECT) to MAINT 193:

sploud ipcs object

```
DMSWSL409I Loading IPCS OBJECT to MAINT 193

Ready;
```

- Use the System Product Editor (XEDIT) to edit CUSTOMER PROFILE:

access 193 r

xedit customer profile

This file contains all of the standard fields that you need to submit APARs (Authorized Program Analysis Reports). As loaded from the product tape, these fields contain sample data to show the type of information required.

Note: The following is a sample CUSTOMER PROFILE file.

```
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. PS Rep = PSR 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. = 012-345-6789   Tie Line = 8-123-7890
------------------------------------------------------------------------
```

- 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

Note: *For information about using IPCS, refer to the* **VM Diagnosis Guide**.

Step 11. Save CMS

Overview

- Save the CMS nucleus
- Enlarge the CMS nucleus to contain the Y-disk directory (if required)

Procedure

■ Save the CMS nucleus:

ipl 190 parm savesys cms

This command saves the CMS nucleus using the name, size, and location parameters defined in DMKSNT.

```
SYSTEM SAVED
```

```
DMSEXG327I The installation DCSS could not be loaded
```

The installation segment is not loaded and saved until after you complete the base installation.

This message does not appear if you indicated in DMSNGP, or in response to the DMSINQ296R prompt (see page 4-16), that you do not want to use the segment.

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

ENTER

Enlarging the CMS Nucleus to Contain the Y-STAT

If you receive the message

```
DMSACC723I Y (19E) R/O
```

the next time you issue the **ipl cms** command, the Y-STAT (Y-disk directory) could not fit within the CMS nucleus that you just saved. In that case, every user who IPLs CMS gets the Y-disk (19E) accessed in non-shared user free space, which is less efficient. You may want to enlarge your CMS nucleus to contain the Y-STAT. See Appendix F, "Enlarging the CMS Nucleus To Contain the Y-Disk Directory (Y-STAT)" on page F-1.

Step 12. 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 2-5.

- Format and allocate the GCS volume (if required)
- 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)
- Install additional GCS systems (optional)

Procedure

Note: If the GCS minidisks already exist, you can skip to page 4-46 and load the GCS code.

- If you are installing GCS on its own volume, use the IPL FMT program to CP format the volume:
  - To use an IPLable version of the program located on tape, mount the tape and IPL the tape until the program is loaded.
  - To use an IPLable version of the program located on disk (the version supplied on your 190 system disk, or one that you placed on some other disk), attach and access the disk, then use the procedure shown below to punch the program to your reader and IPL it into your virtual machine.

Punching the IPL FMT Program from Disk

Issue the following commands:

```
purge rdr class i
close rdr
close pun
spool pun * class i
pun ipl fmt x (noheader

spool rdr class i
ipl 00c clear
```

x is the filemode of the disk where IPL FMT is located.

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

format

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

rdev or ***vdev***

At first level, if you loaded the program from tape, enter the **real** address of the volume; if you punched the program from disk, enter the **virtual** address at which you attached the volume. At second level, enter the **first** level **virtual** address of the volume, which functions as a "real" address.

```
ENTER DEVICE TYPE:
```

devtype

```
ENTER START  PAGE (XXXXX)|CYLINDER (XXX OR XXXX)  OR "LABEL":
```

ENTER

```
ENTER END  PAGE (XXXXX)|CYLINDER (XXX OR XXXX):
```

ENTER

```
ENTER DEVICE LABEL:
```

spsys

Use **spsys** or some other temporary name not listed in the SYSOWN macro in the DMKSYS file.

```
FORMAT STARTED
```

Warning: Do not make ANY entry until you receive the **FORMAT DONE** message, or you may cause the system to issue the message before the formatting is complete.

```
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

If this message indicates **any** errors, stop the installation procedure and contact your Systems Engineer or hardware service personnel. (You may have to initialize the volume with the Device Support Facility. Refer to the *Device Support Facility User's Guide and Reference*.)

```
ENTER FORMAT OR ALLOCATE:
```

- Use the ITASK EXEC to allocate the GCS volume (the volume label must be listed in the SYSOWN macro in DMKSYS).

#cp ipl cms

This command ends the IPL FMT program and gives you access to ITASK.

VM/SP Release *n mm/dd/yy hh:mm:ss*

Or your own version identification.

ENTER

itask allocate

WHAT IS THE REAL ADDRESS OF YOUR *volid* **VOLUME?**
TYPE: REAL ADDRESS OR SKIP

You are prompted for each volume listed in the SYSOWN macro in DMKSYS. You may not be using some of these volumes in your system. Refer to your "Installation Reference Worksheet" on page 2-13 for the labels and addresses of these volumes.

skip or ***rdev*** or ***vdev***

Enter **skip** for all volumes except the one that you are using for GCS.

At first level, enter the **real** address of the GCS volume; at second level, enter the **first** level **virtual** address, which functions as a "real" address at second level.

DASD *nnn* **ATTACH TO MAINT** *nnn*

This message appears only for the GCS volume address that you entered at the previous prompt.

.
.
.

PUN FILE *nnn* **TO MAINT COPY** *nnn* **NOHOLD**

This message appears only once, after you have completed the previous sequence for each volume listed in the SYSOWN macro. It indicates that the IPL FMT program has been punched to MAINT.

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

FORMAT

.
.
.

Note: The system displays the format and allocate sequences for the volume that you are allocating. During this process, the volume is assigned its permanent label, which is obtained from the $ALLOC$ userid in the system directory.

```
ALLOCATION RESULTS
PERM     nnnnn   nnnnn
TEMP     nnnnn   nnnnn
PERM     nnnnn   nnnnn
DEVICE nnn VOLUME volid   ALLOCATION ENDED

ENTER FORMAT OR ALLOCATE:
```

This is the format of the allocation results on a 3380 device. Allocation results on other devices may vary.

Note: This volume becomes "CP-owned" the next time you shut down and reIPL.

#cp ipl cms

This command ends the IPL FMT program.

- Load the GCS code:

spload gcs interface

This command loads the GCS INTERFACE tape file to MAINT 193. These files provide an interface between GCS and IPCS.

```
DMSWSL409I Loading GCS INTERFACE to MAINT 193
```

spload gcs object

This command loads the GCS OBJECT tape file to MAINT 595. This tape file contains the GCS object code, plus the GCSLOAD EXEC and GROUP EXEC files. If MAINT 595 has not been formatted, SPLOAD formats the disk before continuing with the load.

```
DMSWSL409I Loading GCS OBJECT to MAINT 595
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 4-55.*

2. *If you do not have a full screen display device, you can not use GROUP EXEC to create the configuration file, because you can not display the panels. Therefore, you can skip this operation. Refer to*

the **VM/SP Group Control System Command and Macro Reference** *and use the build macros described there to build the configuration file manually. Then continue this step with the nucleus build operation on page 4-55*

access 595 b

copyfile csimes txtameng b csiume = a

This command creates a temporary GCS message file (containing the messages for GROUP EXEC) that has the filename 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 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 2-5.

group *systemname*

This command assigns ***systemname*** as the filename of the GCS configuration file that you are creating 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 (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.

KEY	FUNCTION
PF1 HELP	Shows information pertaining to 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've entered information, but haven't saved 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 you've 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 you've entered on the panel also exists somewhere else (in a directory entry, for instance). If you press PF9, 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, 10, 11	Not Used
ENTER	Saves information you've 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.

Figure 4-3. Function Keys Used with the GROUP EXEC Panels

```
 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

====>
```

Figure 4-4. GROUP EXEC Primary Option Menu

- To complete the Primary Option Menu:

  1. **Fill in or change the SYSTEM NAME.**
     If you invoked GROUP with the ***systemname*** operand, this panel appears with the SYSTEM NAME already filled in. Make sure that the SYSTEM NAME you provide matches the SYSNAME entry for this GCS system in the system name table (DMKSNT).

  2. **Verify your input before selecting another screen.**
     Press PF9 to verify that the SYSTEM NAME you entered has a valid entry in your system name table. You receive two messages: (1) tells whether the SYSTEM NAME exists and (2) says: **Enter selection number 1, 2, or 3.**

     Note: If a configuration file for this SYSTEM NAME already exists (for example, MYGCS GROUP), GROUP EXEC makes a backup copy of it, with a different filetype, before you start making changes:

     MYGCS OLDGROUP (original file, new filetype)
     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   OF
------------------------------------------------------------------------------


        To ADD,    fill in the blanks with the authorized VM userids.
        To CHANGE, type a new userid over the userid 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

====>
```

Figure 4-5. GROUP EXEC Authorized Userids Panel

- To complete the Authorized Userids panel:

1. **Type in the userid(s).**
   Press ENTER or PF6 after typing each userid to save your information and advance to the next space at the same time. Every time you press ENTER or PF6, GROUP EXEC tells you how many userids it has processed. Use PF7 and PF8 if you have more than one screenful of userids.

2. **Verify your input.**
   Press PF9. GROUP EXEC tells you how many userids on the page being viewed are valid or invalid and places arrows beside any userids not found in your VM/SP 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 userids the EXEC has 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

====>
```

Figure 4-6. GROUP EXEC Saved System Information Panel 1

- To 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; 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 userid. GROUP EXEC tells you whether that userid is valid, but it does not prevent you from saving an invalid entry.

  3. **Specify a trace table size.**
     By default, 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 userids have valid entries in your VM/SP directory. If your dump receiver also appears on the other panel of Authorized VM userids, GROUP EXEC verifies it as a "Valid Authorized" userid. Make sure any new SYSTEM DISK or TRACE TABLE information is correct; 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

====>
```

Figure 4-7. GROUP EXEC Saved System Information Panel 2

- To complete the second Saved System Information panel:

  1. **Specify a MAXIMUM NUMBER.**
     Simply type a value of "1" or more, and press ENTER. 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. 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.

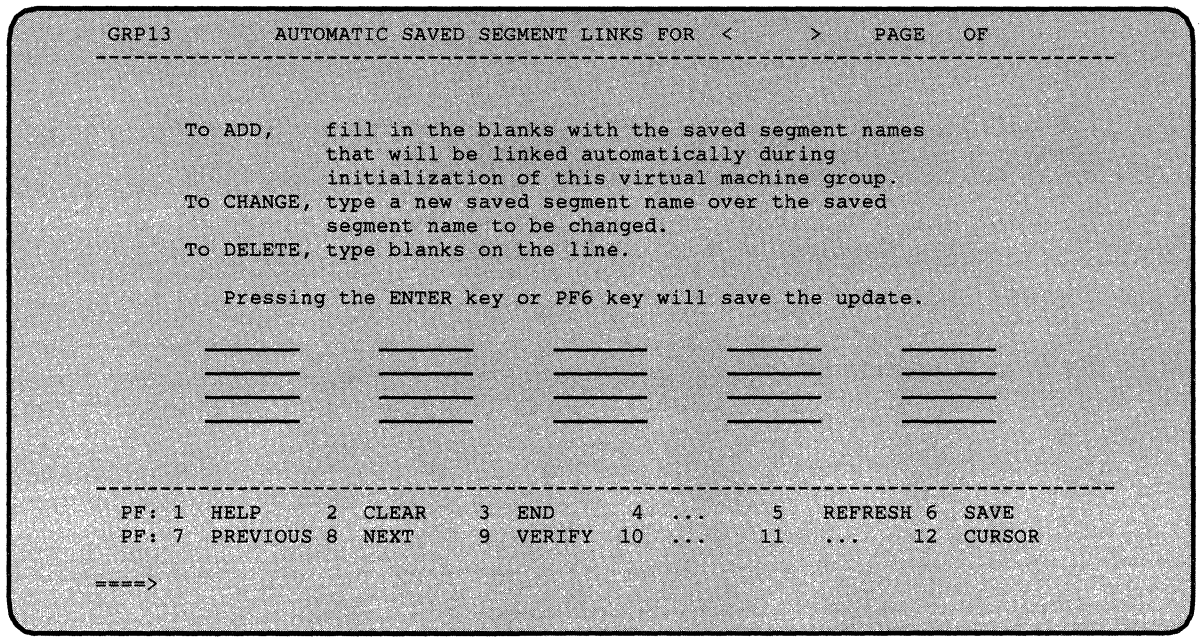


Figure 4-8. GROUP EXEC Automatic Saved Segment Links Panel

- To complete the Automatic Saved Segment Links panel:

  Note: Do not include CMSVSAM and CMSBAM segments here. These VSAM segments are described in 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 ENTER or PF6, the EXEC tells you how many segment names it has processed. Use PF7 and PF8 if you have more than one screen of names.

  2. **Verify your input.**
     Press PF9. 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.

- Once you have provided the necessary input on all panels and returned to the Primary Option Menu for the last time, press PF3 to exit from GROUP EXEC.

- Remove the temporary GCS message file from user storage and erase it from your A-disk:

set language ameng (delete csi user

erase csiume txtameng a

- Invoke ITASK EXEC with the BUILD GCS operands to build and save the GCS nucleus.

  ITASK performs the following actions:

  - Modifies a copy of the GCS loadlist (GCSLOAD EXEC) and updates the entry that contains the filename of the GCS configuration file (the default is GCS) to match the filename of the configuration file that you just created (the default is also GCS), then files the modified loadlist on MAINT 295.

  - Renames the filetype of the GCS configuration file from GROUP to ASSEMBLE.

  - Invokes the SPGEN ASSEMBLE procedure to assemble the configuration file.

  - Invokes the SPGEN NUCLEUS procedure to build and save the GCS nucleus.

itask build gcs *systemname*

systemname is the filename 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
DMSGLB002W File DMKSP MACLIB not found
ASMBLING systemname
.
.
.
```

Note: If you receive an assembly error, you may have to reaccess the following minidisks: **access 295 a** *(contains tailorable files),* **access 595 b** *(contains GROUP EXEC),* **access 191 c** *(contains SPGEN PROFILE),* **access 193 d** *(contains SPGEN EXEC). Then go back and check your configuration file (which ITASK has renamed to* ***systemname*** *ASSEMBLE). For information about the required fields, see "Planning for the Group Control*

System (GCS)" on page 2-5. Correct the file or go through the GROUP EXEC panels again to recreate the file. (If you recreate the configuration file, you must use the same ***systemname****. After you recreate the configuration file, you must rename or erase the old ASSEMBLE file and then rename the filetype of the new configuration file from GROUP to ASSEMBLE.) Issue the* **spgen assemble** ***systemname*** **gcs** *command to assemble the configuration file, then issue the* **spgen nucleus gcs** *command to build and save the nucleus.*

.
.
.

```
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
CP entered;  disabled wait PSW '00020000 00000000'
```

- IPL CMS:

```
ipl cms
VM/SP Release n mm/dd/yy hh:mm:ss
ENTER
Ready;
```

Or your own version identification.

Saving and Printing the GCS Load Map

The GCS load map has been sent to MAINT's virtual reader. To save the load map on disk, issue the following commands:

access 595 e
receive *spoolid* **gcsnuc map e**

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

print gcsnuc map e

Installing Multiple GCS Systems

If you defined multiple GCS systems in DMKSNT and the system directory, perform the following sequence to install each additional nucleus:

1. Access the following minidisks:

   access 295 a
   access 595 b
   access 191 c
   access 193 d

2. (Full screen display devices only) Set up the GROUP EXEC messages in storage (see page 4-46).

3. Create a configuration file for this nucleus (see page 4-47).

4. (Full screen display devices only) Remove the temporary GCS message file (see page 4-55).

5. Edit the GCS loadlist (GCSLOAD EXEC) and change the configuration file entry (`&1 &2 &3` *fn*, where *fn* is the filename of the configuration file for the previously generated GCS nucleus) to specify the filename of the configuration file for this nucleus.

6. Rename the filetype of the configuration file from GROUP to ASSEMBLE.

7. Issue the **spgen assemble *systemname* gcs** command to assemble the configuration file.

8. Issue the **spgen nucleus gcs** command to build and save the new GCS nucleus.

9. Save or print the GCS load map (optional).

Step 13. Load TSAF (Optional)

Overview

The TSAF (Transparent Services Access Facility) component is shipped as a ready-to-run system. Load the TSAF interface files and object code. No build is required.

Procedure

- Invoke ITASK EXEC with the LOAD TSAF operands:

itask load tsaf

ITASK invokes SPLOAD EXEC to do the load, using the information contained in SPLOAD PROFILE. The sample SPLOAD PROFILE directs SPLOAD EXEC to load the TSAF INTERFACE tape file (containing files that enable TSAF to interface with IPCS) to MAINT 193 and load the TSAF OBJECT tape file to MAINT 492.

```
DMSWSL409I Loading TSAF INTERFACE to MAINT 193
```

Note: If MAINT 492 has not been formatted yet, SPLOAD formats the minidisk before continuing with the load.

```
DMSWSL409I Loading TSAF OBJECT to MAINT 492
Ready;
```

Related Procedures

You have now completed your base VM/SP installation. However, there are several related procedures that you may also want to complete:

- **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 discontiguous saved segments (DCSS).**

  For procedures to install the CMSDOS, CMSBAM, CMSVSAM, CMSAMS, and CMSINST segments, see Chapter 6, "Installing Discontiguous 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, refer to *VM System Facilities for Programming*.

- **Install optional feature program products.**

  If you ordered optional feature program products, see Chapter 8, "Installing Optional Feature Program Products on VM/SP" on page 8-1.

- **Apply Service.**

  If a PUT (Program Update Tape) was included with your shipment, you may want to apply some or all of the service updates that it contains. For information about applying preventive and/or corrective service, see "Part 3. VM/SP Service."

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 via XEDIT command
- Copying files
- Printing and punching CMS files
- Formatting disks
- 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 later 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 (supplied with the Starter System) defined in your 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 virtual read/write disk accessed as the A-disk (usually 191)
- 16M of virtual storage for IVPM1; 3M for IVPM2.

Variations

Executing the IVP with Virtual Machines Other Than IVPM1 and IVPM2

1. Specify **userid1** when you issue the **ivp 1** command (on page 5-5).

2. Specify **userid2** when you issue the **ivp 2** command (on page 5-7).

userid1 and **userid2**, respectively, identify the two virtual machines in which the IVP 1 and IVP 2 EXEC procedures execute.

Executing the IVP in a Single Virtual Machine (Single Device)

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 issue the following commands:

ipl cms

```
VM/SP Release n 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 disk.

```
Enter read password:
```

password

Enter the minidisk read 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 device as IVPM1:

logon ivpm1

```
Enter password:
```

password

Enter the password that you defined for this userid in the system 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 n 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, perform the following sequence:

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 disk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the disk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, refer to 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 disk in the CP directory.

End of Formatting the IVPM1 191 Minidisk

ENTER

This entry is required only if you DID NOT format the IVPM1 191 minidisk.

- Invoke the first IVP procedure:

access 193 t

The IVP tests are on this disk.

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 password that you defined for this userid in the system directory during the installation procedure.

```
LOGON AT hh:mm:ss EST day mm/dd/yy
```

- IPL CMS:

ipl cms

```
VM/SP Release n 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 disk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the disk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, refer to 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 disk in the CP directory.

End of Formatting the IVPM2 191 Minidisk

ENTER

This entry is required only if you DID NOT format the IVPM2 191 minidisk.

- Invoke the second IVP procedure:

access 193 t

The IVP tests are on this disk.

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 userid 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.

At second level, press the **PA1** key and issue the following command:

#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 userid.

```
DMKCPI966I Initialization complete.
.
.
.
```

Single Virtual Machine Procedure

- Log on to the selected virtual machine.
- IPL CMS:

ipl cms

VM/SP Release *n mm/dd/yy hh:mm:ss* Or your own version identification.

ENTER

- LINK and ACCESS the MAINT 193 minidisk:

link maint 193 193 rr The IVP EXECs are located on this disk.

Enter read password:

password Enter the minidisk read 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 Discontiguous Saved Segments

Overview

Discontiguous saved segments (DCSSs) are areas of virtual storage outside the address range of a virtual machine. These segments:

- Were previously loaded and saved
- Have names associated with them
- Contain read-only data or reentrant code
- Can be shared by multiple users.

Discontiguous saved segments provide an efficient means of fetching programs by merely connecting the segments to a virtual machine's address space.

Defining Discontiguous Saved Segments

The system name table (DMKSNT ASSEMBLE) defines the name, size, and virtual storage location of all discontiguous saved segments. The sample DMKSNT file shipped on the VM/SP Product Tape contains entries for discontiguous saved segments named CMSVSAM, CMSAMS, CMSDOS, CMSBAM, CMSINST, and HELP and may include entries for DCSSs required by optional feature program products. DMKSNT also defines named saved systems, such as CMS and GCS. A named saved system is a system saved on disk that can be IPLed by name.

A DCSS or named saved system consists of one or more 64K segments. These segments may be shared or non-shared. Shared segments contain the information that is shared by all virtual machines that attach the DCSS or IPL the saved system. Non-shared segments usually contain non-reentrant code -- information that can not be shared because it is unique to each virtual machine that attaches the DCSS or IPLs the saved system.

For more information about discontiguous saved segments, named saved systems, and the basic structure of DMKSNT entries, refer to the *VM/SP Planning Guide and Reference.*

Throughout the following discussion, it will be helpful for you to refer to Figure 6-1 on page 6-3, which shows the suggested layout of discontiguous saved segments and named saved systems in virtual storage. To determine the actual load addresses for your installation, check the entries coded in your DMKSNT file.

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 segment overlays another segment in storage unless the segments are NEVER used at the same time.

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.

To verify your changes, you may want to use SNTMAP EXEC, an SNT mapping tool that resides on the MAINT 193 minidisk. 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 discontiguous saved segments in virtual memory.

For more information about SNTMAP EXEC, see "SNTMAP EXEC" on page 15-21 or refer to the *VM/SP Planning Guide and Reference.*

Loading and Saving Discontiguous Saved Segments

Before a DCSS can be attached or detached by name, it must be loaded and saved at an address that is slightly beyond the highest address of any virtual machine that will attach it. A DCSS must not overlay the defined virtual machine or any other DCSS attached at the same time. However, an unnecessarily high load address may waste real storage, because CP maintains segment table entries for each 1024K (1M) of storage.

After a named segment has been loaded at the correct address, storage protection is assigned with the SETKEY command. Then the segment is saved with the SAVESYS command. For more information about the SETKEY command, refer to *VM/SP CP for System Programming.* For more information about the SAVESYS command, refer to the *VM/SP Planning Guide and Reference.*

Procedures for loading and saving CMSDOS, CMSBAM, CMSVSAM, CMSAMS, and CMSINST are included in this chapter. Installation of the HELP segment is part of the VM/SP base installation procedure. Procedures for installing segments required by optional feature program products are described in Part 2 of this book or in the program product documentation.

VM/SP provides the following tools to help you install the CMS discontiguous saved segments:

- DOSGEN EXEC for the CMSDOS segment
- SAMGEN EXEC for the CMSBAM segment
- VSAMGEN EXEC for the CMSVSAM and CMSAMS segments
- DCSSGEN MODULE for the CMSINST segment.

These tools are located on the MAINT 193 minidisk. You must access this disk **after** any intermediate IPL of CMS.

Suggested Storage Layout

Figure 6-1 shows the storage layout defined by the sample DMKSNT.

Note: Refer to the latest VM/SP Memo to Users for any changes to this layout.

Location	Name	Function
16384K		End of Storage
15296K	CMS	Contains the CMS nucleus code for the file system, device I/O interrupt handlers, debug, etc., and the S- and Y-disk directories. EF0000[1] 3824[2] 239-255[3]
14848K		Inter-Segment Gap
14656K	CMSINST	Contains EXECs and System Product Editor macros. E50000 3664 229-232
14400K	HELP	Contains HELP file directory information. E10000 3600 225-228
14336K	CMSDOS	Contains VSE simulation routines. E00000 3584 224
13504K		Inter-Segment Gap
13312K	CMSBAM	Contains VSE routines. D00000 3328 208-210
12864K	CMSVSAM	Contains CMS/VSAM support. Segments 201-206 are shared; segment 207 is non-shared. C90000 3216 201-207

Figure 6-1 (Part 1 of 2). Suggested Virtual Storage Layout for Discontiguous Saved Segments and Named Saved Systems

Location	Name	Function		
12288K	CMSAMS	Contains CMS/AMS support. Segments 192-197 are shared; segments 198-200 are non-shared. C00000	3072	192-200
5120K		Inter-Segment Gap		
4096K	GCS	Contains GCS supervisor code and common storage. 400000	1024	64-79
0K		CMS virtual machine area. 000000	0	0-63

Figure 6-1 (Part 2 of 2). Suggested Virtual Storage Layout for Discontiguous Saved Segments and Named Saved Systems

[1]Hex load address
[2]Starting page number
[3]Segment numbers

Procedures

This section contains recommended installation procedures for the CMSDOS, CMSBAM, CMSVSAM, CMSAMS and CMSINST discontiguous saved segments.

These segments have the following general installation requirements:

- Generate your VM/SP system with enough reserve DASD space to contain the segments that you want to install.
- Install the segments in the order indicated.

If there are other discontiguous saved segments that you want to install:

- Make sure that the segments are properly defined in DMKSNT.
- Use the following procedures as guidelines.

CMSDOS

Installing CMSDOS

Overview

The CMSDOS segment contains text files needed to create a CMS/DOS environment that simulates DOS/VSE (Disk Operating System/Virtual Storage Extended) under CMS.

To install CMSDOS, use the DOSGEN EXEC procedure. You invoke DOSGEN with the hex load address and the name that you want to assign to this segment. This load address and name must match the address and name coded in DMKSNT.

Notes:

1. *The CMSDOS segment must be defined at a higher address than the CMSBAM, CMSVSAM, and CMSAMS segments.*

2. *Before you load and save CMSDOS, you must define your virtual machine with at least 512K free storage above the end of the CMSDOS segment. This provides room for the loader tables, which occupy the top pages of virtual storage. After you load and save the segment, the loader tables and the 512K free storage are no longer required.*

DOSGEN performs the following operations:

- 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 accessed A-disk 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 the A-disk as LOAD MAP A5.

Procedure

- Log on as MAINT (if you have not already done so). The default size of MAINT's virtual machine is 16M. If you changed the default, you must define your storage as 16M.
- Set up your system:

ipl 190

VM/SP Release *n mm/dd/yy hh:mm:ss* — Or your own version identification.

access (noprof — This command suppresses execution of MAINT's PROFILE EXEC.

ENTER

access 193 t — DOSGEN EXEC is on this disk.

set emsg on — This command sets all error messages on.

- Invoke DOSGEN EXEC:

dosgen e00000 cmsdos — This address and name must match the address and name coded in DMKSNT. This example uses the address shown in Figure 6-1 on page 6-3.

Messages

If DOSGEN detects an error in the address that you specified:

DMSGEN095E INVALID ADDRESS

If DOSGEN can not find a read/write accessed A-disk:

DMSGEN006E NO READ/WRITE A-DISK ACCESSED.

If DOSGEN finds unresolved external references while loading the text files:

DMSGEN111E DOSGEN FAILED DUE TO LOAD ERRORS.

If DOSGEN detects an error while assigning the storage key or saving the segment:

DMSGEN412S DOSGEN FAILED DUE TO SETKEY ERRORS

DMSGEN141S DOSGEN FAILED DUE TO SAVESYS ERRORS

```
SYSTEM SAVED
DMSGEN715I DOSGEN COMPLETE
```

- If you want to save the load map (LOAD MAP), rename it and copy it from the A-disk to the 193 minidisk (which you accessed as T). For example:

copy load map a cmsdos segmap t

Installing CMSBAM

Overview

Note: The CMSDOS segment must already be installed.

The CMSBAM segment 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).

To install CMSBAM, use the SAMGEN EXEC procedure.

Before you invoke SAMGEN EXEC, you need to 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, you must not define your storage larger than the address at which you loaded CMSDOS.

SAMGEN EXEC performs the following operations:

- Fetches the simulated VSE phases from the CMSBAM DOSLIB file, which is supplied as part of VM/SP.
- Loads the simulated phases at the designated address.
- Assigns a storage protection key of X'F'.
- Saves the segment.

Procedure

- Log on as MAINT (if you have not already done so).
- Install CMSDOS (if that segment is not already installed).
- Define your virtual machine:

define storage 14M

This example follows Figure 6-1 on page 6-3. **Do not define your virtual storage larger than the address at which you loaded the CMSDOS segment.**

```
STORAGE=14336K
CP entered;  disabled wait PSW '00020000 00000000'
```

ipl cms

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

ENTER

access 193 t

SAMGEN EXEC is on this disk.

set sysname cmsbam *xxxxxxxx*

This command changes the CMSBAM segment's name entry in the CMS SYSNAME table to ***xxxxxxxx***. Enter any 1-to-8-character string that is NOT used as the name of a segment in DMKSNT. For example, you could enter: **set sysname cmsbam test.**

- Make sure that you have the following CMS files (distributed as part of VM/SP) by issuing a listfile command for each one:

  SAMGEN EXEC
  CMSBAM DOSLIB

  These files should be located on the MAINT 193 minidisk.

- Invoke SAMGEN EXEC:

set dos on

This command places the CMS virtual machine in a CMS/DOS environment.

samgen

```
DMSSGN363R ENTER LOCATION WHERE CMSBAM WILL BE LOADED AND SAVED
```

d00000

This address must match the address coded in DMKSNT. This example uses the address shown in Figure 6-1 on page 6-3.

```
DMSSGN364I FETCHING CMSBAM...
.
.
.
DMSSGN366R ENTER NAME OF SYSTEM TO BE SAVED
```

cmsbam

This name must match the name coded in DMKSNT.

```
SYSTEM SAVED
DMSSGN365I SYSTEM CMSBAM SAVED
```

CMSVSAM/CMSAMS

Installing CMSVSAM and CMSAMS

Overview

Note: The CMSDOS and CMSBAM segments must already be installed.

The CMSVSAM and CMSAMS segments allow you to use Virtual Storage Access Method (VSAM) and Access Method Services (AMS) under CMS.

To install CMSVSAM and CMSAMS, use the VSAMGEN EXEC procedure.

Before you can invoke 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 segments plus at least one segment to contain the CMS control blocks and free storage used during the installation process.
- Access in read/write mode an A-disk that is large enough to contain all of the VSAM and AMS text decks required to generate the shared segments. The amount of space required depends on the minidisk device type, as shown below:

Device Type	Cylinders/Blocks
3330	30
3340	70
3350	15
3375	22
3380	14
FBA	13440

You can invoke 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 the A-disk.	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.

Procedure

- Log on as MAINT (if you have not already done so).
- Install the CMSDOS and CMSBAM segments (if those segments are not already installed).
- Access an A-disk of the required size.
- 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 may have to do it for you.)
- Define your virtual machine (if you have not already done so):

define storage 14m

This example follows Figure 6-1 on page 6-3. **Do not define your virtual storage larger than the address at which you loaded the CMSDOS segment.**

```
STORAGE=14336K
CP entered;  disabled wait PSW '00020000 00000000'
```

ipl cms

```
VM/SP Release n mm/dd/yy hh:mm:ss
```

Or your own version identification.

ENTER

access 193 t

VSAMGEN EXEC is on this disk.

set sysname cmsbam *xxxxxx*

xxxxxx is any character string that is NOT the same as the name of a segment in DMKSNT.

- Invoke VSAMGEN EXEC:

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 A-disk, select function **4**, **5**, or **6**.

If the DOSLIBs currently reside on an accessed disk, select function **7**, **8**, or **9**. In this case, it is not necessary to have the text files available.

Messages

While VSAMGEN is processing, you may receive error and information messages. These messages are self-explanatory. Messages labeled **2101I** are information messages from the linkage editor and may be ignored.

```
DMSVGN363R ENTER LOCATION WHERE CMSVSAM WILL BE LOADED AND SAVED:
```

c90000

This address must match the address coded in DMKSNT. This example uses the address shown in Figure 6-1 on page 6-3.

```
DMSVGN364I FETCHING CMSVSAM ...
.
.
.
DMSVGN366R ENTER NAME OF SYSTEM TO BE SAVED:
```

cmsvsam

This name must match the name coded in DMKSNT.

```
DMSVGN365I SYSTEM CMSVSAM SAVED.
.
.
.
DMSVGN363R ENTER LOCATION WHERE CMSAMS WILL BE LOADED AND SAVED:
```

c00000

This address must match the address coded in DMKSNT. This example uses the address shown in Figure 6-1 on page 6-3.

```
DMSVGN364I FETCHING CMSAMS ...
.
.
.
DMSVGN366R ENTER NAME OF SYSTEM TO BE SAVED:
```

cmsams

This name must match the name coded in DMKSNT.

```
DMSVGN365I SYSTEM CMSAMS SAVED
.
.
.
```

- If you want to save CMSAMS DOSLIB and CMSVSAM DOSLIB, you must rename and copy them now.

CMSINST

Installing CMSINST

Overview

The installation segment (CMSINST is the default name) is designed to contain EXECs and System Product Editor (XEDIT) macros. When a frequently-used EXEC or Editor macro resides in a discontiguous saved segment, multiple users can share the same executing copy.

To install CMSINST, use the DCSSGEN command procedure. This procedure lets you build, load, and save a CMSINST segment that contains the EXECs and Editor macros that you select for your installation.

Before you issue the DCSSGEN command, you must create a loadlist file that contains a list of the EXECs and System Product Editor macros to be loaded into the CMSINST segment. **The loadlist must be a fixed-format file with a logical record length of 80.** Each record in the file must contain the fileid of one EXEC or System Product Editor macro or a comment. DCSSGEN processes the records sequentially.

The format of a DCSSGEN loadlist entry is:

```
fn  ft  [fm  [execname  [exectype] ] ]
```

fn
is the filename of the EXEC or Editor macro to be loaded.

ft
is the filetype of the EXEC or Editor macro to be loaded.

fm
is the filemode of the EXEC or Editor macro to be loaded. If the filemode is specified as *, DCSSGEN loads the first file in the disk search order that satisfies the filename and filetype qualifications.

execname
is the filename to be assigned to the loaded EXEC or Editor macro. The default is '=', which means that the present filename is to be used.

exectype
is the filetype to be assigned to the loaded EXEC or Editor macro. The default is '=', which means that the present filetype is to be used.

The filename and filetype 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 may also be from one to eight characters. However, they are not limited to the

filename and filetype 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 may 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
```

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, refer to the *VM/SP CMS Command Reference.*

When you issue the DCSSGEN command, you specify the fileid of your loadlist file. You also specify the name that you want to assign to the discontiguous saved segment. The name of the segment must match the name coded in DMKSNT. The default segment name is CMSINST.

DCSSGEN performs the following operations:

- Processes the loadlist file, sequentially loading each EXEC and Editor macro into storage.
- Saves the segment.
- Writes a load map to the A-disk 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
```

Procedure

- Log on as MAINT (if you have not already done so).
- Define your virtual machine:

define storage 16m

STORAGE=16384K

ipl 190

VM/SP Release *n mm/dd/yy hh:mm:ss* Or your own version identification.

ENTER

access 193 b DCSSGEN MODULE is on this disk.

- Invoke DCSSGEN MODULE:

dcssgen ***dcsslist file a dcssname***

dcsslist file a is the fileid of the file that contains the list of EXECs and Editor macros to be loaded into the segment.

dcssname is the name of the segment. This name must match the name coded in DMKSNT. If you do not specify a segment name, the default name 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 installation segment (the default is YES), then this segment is used each time a user IPLs CMS with virtual storage defined less than the starting address of the segment (14656K). If you previously indicated that you did not want to use the 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.

Messages

While DCSSGEN is processing, you may receive error or warning messages that indicate specific conditions. If errors were encountered, after processing is complete you receive the following prompt:

```
DMSEXG298R An error has been detected while building the DCSS.
           Do you still want the DCSS saved?
           Enter 1 (YES) or 0 (NO).
```

Enter **1** to disregard the error(s) and save the segment, or enter **0** to not save the segment. If you do not save the segment, you receive the message:

DMSEXG288I *dcssname* **not saved**

If DCSSGEN encounters an error while saving the segment, you receive the message:

DMSEXG288E *dcssname* **not saved**

If your virtual machine is not large enough to contain the segment (you need 16M), you receive the message:

DMSEXG284E The DCSS is not completely inside the virtual machine

To correct this situation, increase the size of your virtual machine, re-IPL CMS, and reissue the DCSSGEN command.

Chapter 7. Installing a New System National Language

Overview

VM/SP is shipped with American 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.

Note: A user's virtual machine is set to the system national language during logon, unless the directory entry for that userid contains an OPTION LANG statement that overrides the system national language.

You can install another national language on your system:

- To replace the current system national language. Use the procedure described in this chapter.
- As an option available to selected system users. Refer to *VM System Facilities for Programming*.

VM/SP national languages (except American English) are distributed on national language feature tapes. The files on a national language feature tape contain translated information. The national language is identified by a unique 1-to-5-character *langid*, as shown in Figure 7-1.

Langid	Language
AMENG	American English
KANJI	Kanji (Japan)
UCENG	Uppercase English
PORTG	Brazilian Portuguese
FRANC	French
GER	German

Figure 7-1. National Language langids

To install a language as the new system national language:

- Load the language files from the national language feature tape to minidisks.
- Identify the new language in the system generation profiles and update the CMS translation tables (if necessary).
- Build a new CP nucleus containing the CP language files.
- Build and save a new CMS nucleus containing the CMS language files.
- Shut down the system and do a warm start to bring the new CP and CMS nuclei on line.
- (If GCS is installed) Create a new GCS configuration file (optional).
- (If GCS is installed) Build and save a new GCS nucleus containing the GCS language files.

Contents of the National Language Feature Tape

A national language feature tape contains two types of files:

- Source files
- Object code files.

Figure 7-2 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

Figure 7-2. National Language Feature Tape File Layout

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.

Fileid	*Description*
DMKMES[*y*] REPOS	CP message repository, containing the translated versions of CP system messages.
DMSMES[*y*] REPOS	CMS message repository, containing the translated versions of CMS system messages.

DMSSPA[*y*] DLCS — CMS command syntax file, containing the translated syntax definitions for CMS commands.

DMSTRT[*y*] ASSEMBLE — CMS uppercase translate table, mapping lowercase alphabetic characters to uppercase for the language.

CSIMES[*y*] REPOS — GCS message repository, containing the translated versions of GCS messages.

ATSCMR[*y*] REPOS — TSAF message repository, containing the translated versions of TSAF messages.

xxxxxxxx HELP*cccc* — HELP files, where *xxxxxxxx* is the message number or the name of the command and *cccc* is the name of the HELP facility component.

HELP files have the same fileids for all languages; therefore, the HELP files for each language installed on your system must reside on a different disk.

In the source filenames, *y* is a "country code" character that identifies the national language and langid. The recognized values for the country code character are shown in Figure 7-3. These values are stored in a file called VMFNLS LANGLIST.

Note: The American English versions of translatable source files have six-character filenames; they do not use the country code character.

Country Code (y)	Language	langid
A	Kanji (Japan)	KANJI
B	Uppercase English	UCENG
C	Brazilian Portuguese	PORTG
D	French	FRANC
E	German	GER

Figure 7-3. National Language Country Codes

Object Files

National language object files have been converted to machine-readable form. These files are used to build the new CP, CMS, or GCS nucleus during the installation procedure. The text file produced for the TSAF message repository is loaded into the TSAF virtual machine storage using the SET LANGUAGE command.

Fileid	*Description*
DMKMES TXT*langid*	CP message repository, created by GENMSG.
DMSMES TXT*langid*	CMS message repository, created by GENMSG.
DMSSPA TXT*langid*	CMS system command syntax definition file, created by CONVERT COMMANDS.
DMSSSY TXT*langid*	CMS system national language translation and synonym table, created by CONVERT COMMANDS.
DMSTRT TXT*langid*	CMS uppercase translation table, created by VMFASM.
CSIMES TXT*langid*	GCS message repository, created by GENMSG.
ATSUME TXT*langid*	TSAF message repository, created by GENMSG.

Procedure

Notes:

1. *Perform this procedure while logged on to the MAINT userid.*
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 A. Load the Language Files from Tape to Disk

- Mount the national language feature tape on device 181.
- Invoke ITASK EXEC with the LOAD LANG *comp* operands:

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 TXT*langid*, REPOS, and HELP*cccc* files. If you do not have all of the VM/SP components installed in your system, issue this command with the **cp**, **cms**, **gcs**, **tsaf**, **help**, or **cmssrc** operand to load the files for each component that you do have installed. See "ITASK EXEC" on page 15-12.

ITASK invokes SPLOAD EXEC to load the language files to the destinations specified in SPLOAD PROFILE, as indicated in Figure 7-4 and Figure 7-5 on page 7-6.

Notes:

1. *REPOS files for source are loaded to the object disks for the corresponding components.*
2. *You are prompted for the address of the disk where you want the HELP files loaded.*

Source File	Minidisk
DMKMES*y* REPOS	MAINT 194
DMSMES*y* REPOS	MAINT 193
DMSSPA*y* CSDL	MAINT 393
DMSTRT*y* ASSEMBLE	MAINT 393
CSIMES*y* REPOS	MAINT 595
ATSCMR*y* REPOS	MAINT 492
xxxxxxxx HELP*cccc*	MAINT *vdev* (unique HELP disk for this language)

Figure 7-4. National Language Source File Minidisk Destinations

Object File	Minidisk
DMKMES TXT*langid*	MAINT 194
DMSMES TXT*langid*	MAINT 193
DMSSPA TXT*langid*	MAINT 193
DMSSSY TXT*langid*	MAINT 193
DMSTRT TXT*langid*	MAINT 193
CSIMES TXT*langid*	MAINT 595
ATSUME TXT*langid*	MAINT 492

Figure 7-5. National Language Object File Minidisk Destinations

Step B. Identify the New Language in the System Generation Profiles and Update the CMS Translation Tables (If Necessary)

- Modify SPGEN PROFILE:

xedit spgen profile

.
.
.

The CP_lang, CMS_lang, and GCS_lang statements in SPGEN PROFILE identify the langid of the system national language for each nucleus. Change these statements to specify the langid of the language that you are installing. **Make sure that you specify the SAME language in each statement.**

file

- Modify the CMS nucleus generation profile (DMSNGP ASSEMBLE):

xedit dmsngp assemble

.
.
.

Change the LANGID statement to identify the new system national language.

If the new language contains Double-Byte Character Set data, change the DBCS statement to YES.

file

spgen assemble dmsngp cms

Updating the CMS Translation Tables

The DMSTRT*y* ASSEMBLE file that you loaded from the language feature tape in Step A contains read-only system translation tables used by other CMS modules for:

- Uppercase translation
- 327x workstation support (displayable characters, APL, TEXT, etc.).

You may 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 DCSS 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*y* using the XEDIT command with the UPDATE option:

```
access 393 c
xedit dmstrty assemble c (update
```

This option creates a file named DMSTRT*y* UPDATE that contains your changes; the source file is not changed. For more information about using the XEDIT command with the UPDATE option, refer to the *VM/SP CMS Command Reference* 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 "Files for Updating Source" on page 12-4.

Finally, use VMFNLS EXEC to generate the associated object file (DMSTRT TXT*langid*) that includes your changes:

```
vmfnls dmstrty assemble dmssp
```

For more information about VMFNLS EXEC, see page 15-67.

Step C. Build a New CP Nucleus Containing the CP Language Files

- Invoke SPGEN EXEC with the NUCLEUS CP operands:

spgen nucleus cp	SPGEN issues the VMFLOAD command to build the nucleus, using the loadlist, control file, and langid specified in SPGEN PROFILE.
.	
.	
.	
NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=*nnn* , LAST CYL/BLK USED=*nnn*	This 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, etc., or unpredictable results may occur. An easy way to check this is to look at VMUSERS DISKMAP (which you created after you tailored the directory) and make sure that **LAST CYL/BLK USED=*nnn***, as displayed above, does not exceed the $CPNUC$ ending cylinder/block defined in the directory. To look at this file, you must first issue the **#cp ipl 190** command.
CP entered; disabled wait PSW '00020000 00000012'	This is the message that you receive if the nucleus loaded correctly. However, on many processors this message does not display automatically. If the PSW message does not display within about one minute, enter **#cp display psw**.

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 disk, issue the following commands:

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

After **VM READ** appears in the corner of your screen, issue:

```
ENTER
access 194 c
query rdr
```

MAINT *spoolid* **M PUN** *nnnnnnnn* **0001 NONE STANDARD**

```
receive spoolid cpnuc map c
```

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

```
print cpnuc map c
```

#cp ipl 190	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 D. Build and Save a New CMS Nucleus Containing the CMS Language Files

- Invoke SPGEN EXEC with the NUCLEUS CMS operands:

spgen nucleus cms	SPGEN issues the VMFLOAD command to build the nucleus, using the loadlist, control file, and langid specified in SPGEN PROFILE.

Nucleus Generation Prompts and Responses

Note: Each of the following prompts appears ONLY if the corresponding statement in DMSNGP is missing or empty (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?).

| **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 =**

| ***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** — The system national language does not use a level id.

| **DMSINQ296R Should the installation segment be used? Enter 1 (YES) or 0 (NO).**

| The installation segment is an optional DCSS into which you can place frequently-used EXECs and System Product Editor 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 installation 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 disk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

ENTER

The default is the address of the system disk.

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn

nnnnn is the cylinder/block location on 190 where the nucleus is written. Enter the correct address for your VMSRES DASD type:

Device	Address
3330	071 cyl
3340	179 cyl
3350	033 cyl
3375	050 cyl
3380	031 cyl
FBA	29184 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 disk 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, refer to the description of the IPL command in the *VM/SP CP 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** *n 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** *n mm/dd/yy hh:mm:ss*	This is the default version identification. If you defined your own version identification, it appears here and each time you IPL CMS.
ENTER	Your CMS system is saved. From now on, instead of IPLing the MAINT 190 system disk, 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 disk, issue the following commands:

```
#cp spool prt nohold
#cp close prt
```

```
PRT FILE spoolid TO MAINT COPY 0001 NOHOLD
```

```
access 193 r
receive spoolid cmsnuc map r
```

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

```
print cmsnuc map r
```

Step E. Shut Down and Do a Warm Start

Note: If you have been using two terminals for the installation, go back to the console (the terminal where the OPERATOR userid is logged on) and do this 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. Be sure to wait until all devices are drained, as indicated by the **Ready** message.

```
Ready;
```

```
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 n, 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 userid.

```
DMKCPJ952I nnnnK system storage
```

For some processors, *nnnn***K** may be less than your actual system storage.

```
.
.
.
DMKCPI966I Initialization complete
```

- Log on to the MAINT userid by doing ONE of the following:

– Issue the **#cp disconn** command to disconnect the OPERATOR userid, 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 n mm/dd/yy hh:mm:ss          Or your own version identification.
ENTER
```

- If you have GCS installed in your system, complete Steps F and G.

Step F. Create a New GCS Configuration File (Optional)

You may want to create a GCS configuration file for this language. For example, you may want to change the authorized userids or add new userids, or you may want to change the text of your system id to appear in the new language.

If you do not want to create a new configuration file, go to "Step G. Build and Save a New GCS Nucleus Containing the GCS Language Files" on page 7-18.

- Access the required minidisks:

```
access 295 a
access 595 b
access 191 c
access 193 d
```

- Set up the GROUP EXEC messages in storage:

Note: If you do not have a full screen display device, you can not use the GROUP EXEC to build the configuration file. Therefore, DO NOT set up the GROUP EXEC messages in storage. Refer to the VM/SP Group Control System Command and Macro Reference and use the build macros described there to build the configuration file manually. Then continue the language installation by modifying the GCS loadlist (see page 7-17).

copyfile csimes txt*langid* b csiume = a

langid 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 GROUP EXEC) that has the filename 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 GROUP EXEC:

group *systemname*

This command assigns ***systemname*** as the filename of the GCS configuration file that you are creating 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 3-102.

If you used the existing system procedure, see page 4-47.

- After you complete the panels and exit from GROUP EXEC, remove the temporary GCS message file from user storage and erase it from your A-disk:

set language *langid* (delete csi user

langid identifies your **current** system national language, not the one you are installing.

erase csiume txt*langid* a

- Modify the GCS loadlist:

copyfile gcsload exec b = = a (replace

This command writes a copy of the loadlist to MAINT 295 and replaces the loadlist used to generate the previous GCS nucleus. DO NOT modify the source file on MAINT 595.

xedit gcsload exec a

.
.
.

Locate the configuration file entry (&1 &2 &3 GCS) in the loadlist and change the filename from the default (GCS) to ***systemname***.

- Rename and assemble the configuration file:

rename *systemname* group a = assemble =

spgen assemble *systemname* gcs

Step G. Build and Save a New GCS Nucleus Containing the GCS Language Files

- Invoke SPGEN EXEC with the NUCLEUS GCS operands:

spgen nucleus gcs

SPGEN issues the VMFLOAD command to build and save the GCS nucleus, using the loadlist, control file, and langid specified in SPGEN PROFILE.

.
.
.

```
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 *n mm/dd/yy hh:mm:ss* Or your own version identification.

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 on disk, issue the following commands:

access 595 e
receive *spoolid* **gcsnuc map e**

Once you have the load map saved on disk, you can print a copy of it by issuing the following command:

print gcsnuc map e

Installing Multiple GCS Systems

If you have multiple GCS systems installed in your system, perform the following sequence to regenerate each nucleus using the new GCS language files:

1. Access the required minidisks (see page 7-16).
2. Set up the GROUP EXEC messages in storage (see page 7-16).
3. Create a configuration file for this nucleus (see page 7-17).
4. Remove the temporary GCS message file (see page 7-17).
5. Modify the GCS loadlist (see page 7-17) to specify the filename of the configuration file for this nucleus.
6. Rename the filetype of the configuration file from GROUP to ASSEMBLE.
7. Issue the **spgen assemble *systemname* gcs** command to assemble the configuration file.
8. Issue the **spgen nucleus gcs** command to build and save the new GCS nucleus.
9. Save or print the GCS load map (optional).

Part 2. Optional Feature Program Product Installation

Chapter 8. Installation of Optional Feature Program Products

Overview

The VM/SP System Offering is a package that includes VM/SP and a predefined set of optional feature program products.

Before you install optional feature program products, use the DIRECGEN EXEC to tailor the CP directory file based on the products you plan to install. And, change logon passwords in the CP directory file with the PASSMOD EXEC. Then, use the INSTFPP EXEC to install the selected optional feature program products from the optional feature product tape.

The following chart outlines the steps for installing optional feature program products:

Formatting and Allocating	• Format and allocate DASD volumes you plan to use during installation.
Before Running DIRECGEN	• Logon to MAINT userid • Make sure DIRMAINT is not operational • Make sure you have read/write access to the input CP directory file and the DIRECGEN EXTENTS file (usually on MAINT 295) • Make sure your CP directory file has fixed length, 80 character records • Make sure the 193 minidisk is linked and accessed • Make sure the 319 minidisk is linked read/write • Make sure you have enough space on the disk containing the input CP directory file for the output CP directory file • Tailor the DIRECGEN EXTENTS file.
Running DIRECGEN	• Invoke DIRECGEN • Select the products you want to install.
After Running DIRECGEN	• Refer to the DIRECGEN HISTORY file • Type in DISKMAP VMUSERS DIRECT to execute DISKMAP EXEC (optional) • Increase the size of existing minidisks if necessary.
Before Running PASSMOD	• Make sure you have read/write access to the CP directory file.
Running PASSMOD	• Invoke PASSMOD • Change the passwords.

Figure 8-1 (Part 1 of 2). Optional Feature Program Product Installation Overview

After Running PASSMOD	• Type in DIRECT VMUSERS to place directory on-line • Logoff • Logon to MAINT userid • Format common minidisks if necessary.
Before Running INSTFPP	• Make sure you are logged on to MAINT userid and MAINT has all privilege classes • Make sure MAINT 319 minidisk is linked read/write • Make sure MAINT 191, MAINT 193, and MAINT 194 are linked • Mount the optional feature product tape • Make sure you have a hardcopy of your CP directory available • Print and review the product Memos to Users • Make sure your virtual storage size is 16M unless otherwise specified by the product Memo to Users.
Running INSTFPP	• Invoke INSTFPP to install the products • Select the products you want to install.
After Running INSTFPP	• Execute manual installation and verification procedures if necessary • Tailor product dependent files if necessary • Resave CMS if necessary • Consider placing file directory information for shared, read/only minidisks into a DCSS.

Figure 8-1 (Part 2 of 2). Optional Feature Program Product Installation Overview

Chapter 9, "Information About Optional Feature Program Products" on page 9-1 contains a list of optional feature program products included on the optional feature product tape and other information related to product installation. Refer to the chapter when needed as you install optional feature program products.

Procedure

Formatting and Allocating DASD Volumes

The installation execs do not format and allocate DASD volumes, such as PROFPK and SQLPK, that you may use when installing optional feature program products.

In addition, the optional feature program product installation execs do not format and allocate additional volumes, such as VMPK02 and VMPK03, that might be available at your installation. The needed control statements for these volumes are not in the VM/SP sample directory.

If additional DASD volumes are needed for product installation, do the following:

1. See the *VM/SP Planning Guide and Reference* for a description of the MDISK directory control statement.
2. Follow the corresponding statements in the VM/SP sample directory for volume VMPK01 as an example when coding these statements.
3. Add an MDISK control statement under the $ALLOC$ userid entry for each additional volume.
4. Add an MDISK statement defining a full-pack minidisk to the userids MAINT and SYSDUMP1 for each of these volumes.

To format and allocate these volumes, enter the following commands from the MAINT userid for each of the volumes you want to format:

Note: If possible, format your DASD at first level.

vary on *rdev*

rdev is the real address of the DASD volume. (At second level, enter the first level virtual address that functions as a second level "real" address.)

```
rdev VARIED ONLINE
Ready;
```

attach *rdev* * *vdev*

rdev is the real address of the DASD volume.
vdev is an available virtual address.

```
DASD rdev ATTACH TO MAINT vdev
Ready;
```

spool punch *
pun ipl fmt s (nohold

From the MAINT userid, load the "IPL FMT" file to your card reader.

```
PUN FILE nnnn TO MAINT COPY 001 NOHOLD
Ready;
```

order reader *nnnn*

nnnn is the number received in the previous system response.

```
0001 FILE ORDERED
Ready;
```

ipl 00c

IPL the card reader to start the FORMAT/ALLOCATE program.

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

format

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

vdev

vdev is the virtual address of the DASD volume.

```
ENTER DEVICE TYPE:
```

devtype

devtype is the device type of the DASD. Enter 3330-11, 3350, 3375, 3380, or FB-512 for the 3370 or 3370-2.

```
ENTER START PAGE (XXXXX)|CYLINDER (XXX OR XXXX) OR "LABEL":
```

ENTER

```
ENTER END PAGE (XXXXX)|CYLINDER (XXX OR XXXX):
```

ENTER

```
ENTER DEVICE LABEL:
```

xxxxxx

xxxxxx is your label name, for example VMPK02.

```
FORMAT STARTED
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

If the previous message **does not** show 000 read-check errors, stop the installation process and contact your Systems Engineer or hardware service representative. (You might have to initialize the volume with the Device Support Facility. Refer to the *Device Support Facility User's' Guide and Reference*, GC35-0033.)

```
ENTER FORMAT OR ALLOCATE:
```

allocate

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

vdev

vdev is the virtual address of the DASD volume.

```
ENTER DEVICE TYPE:
```

devtype

devtype is the device type of the DASD. Enter 3330-11, 3350, 3375, 3380, or FB-512 for the 3370 or 3370-2.

```
ENTER DEVICE LABEL:
```

xxxxxx

xxxxxx is your label name, for example VMPK02.

```
ENTER ALLOCATION DATA FOR VOLUME xxxxxx
TYPE PAGE PAGE|CYL CYL
```

Find the correct start and end page/cylinder values for your DASD type in the following table:

DASD Type	Page / Cylinder Start	End
3330-11	000	807
3350	000	554
3370	00002	69749
3370-2	00002	89093
3375	000	958
3380	000	884
3380-E4	000	1769

Then, using the correct values, invoke the following command:

perm *sssss eeeee*

sssss is the start cylinder or page for DASD type.

eeeee is the end cylinder or page for DASD type.

Note: For this newly allocated space to be properly used in a system, the ending allocation address must be within the formatted range. This is important when running a VM guest operating system.

end

ALLOCATION RESULTS
PERM *xxxxx xxxxx*
DEVICE *vdev* **VOLUME** *xxxxxx* **ALLOCATION ENDED**
ENTER FORMAT OR ALLOCATE:

Do not respond to this message.

At *first level,*

ATTN to enter the CP environment. (For example, press PA1 on a 3278 terminal.)

Warning

If you are running a second level machine, change the first level linend character # to some other character if you have not done so already. Enter TERM LINEND % at first level if you did not do so previously.

At *second level,* enter

#cp ipl cms where # is the linend character of your *second level* system.

detach *vdev*

vdev is the virtual address of the DASD volume.

DASD *rdev* **DETACHED MAINT** *vdev*

attach *rdev* system *xxxxxx*

rdev is the real address of the DASD volume. (At second level, enter the first level virtual address that functions as a second level "real" address.)

xxxxxx is your label name, for example VMPK02.

DASD *rdev* **ATTACH TO SYSTEM** *xxxxxx*

The FORMAT/ALLOCATE step is complete.

Using the DIRECGEN EXEC to Tailor the CP Directory File

DIRECGEN sets up directory entries for optional feature program products you want to install. Execute DIRECGEN after you create the VM/SP Release 5 base system and before you install optional feature program products.

Note: DIRECGEN can define userids and minidisks needed by optional feature program products on DASD types other than the VM/SP system residence volume but only on DASD types supported by VM/SP System Offering.

Before Running DIRECGEN

Before running DIRECGEN, make sure:

- You logon to MAINT userid
- DIRMAINT is not operational
- You have read/write access to the input CP directory file and the DIRECGEN EXTENTS file (usually on MAINT 295)
- Your CP directory file has fixed length, 80-character records
- The 193 minidisk is linked and accessed
- The 319 minidisk is linked read/write
- You have enough space on the disk containing the input CP directory file for the output CP directory file
- You tailor the DIRECGEN EXTENTS file. This file is usually on the MAINT 295 minidisk and is used to identify areas on DASD volumes that should be searched for available minidisk space. Read the comments in this file before making changes.

  Do the following to update this file:

  - Refer to the "Product Directory Information" on page 9-5 to calculate the DASD space needed for the set of products you plan to install.
  - Refer to your directory diskmap to determine where sufficient DASD space is available for the product minidisks.
  - XEDIT the DIRECGEN EXTENTS file and read the comments provided.
  - Change default entries in the file if necessary.

Notes:

1. *You do not need the userid $MDISK$, used in previous releases of VM/SP. Make sure it is not in your input CP directory.*

2. *If you add DASD volumes to the DIRECGEN EXTENTS file, make sure you have made the necessary CP directory updates to reflect the CP reserved areas of the new volume (such as adding a $ALLOC$ minidisk entry).*

3. *List extents for a volume in the DIRECGEN EXTENTS file in sequential order from the beginning of the volume.*

4. *Do not overlap extents for a volume in the DIRECGEN EXTENTS file.*

5. *Plan to select all the products you want to install on your system the first time you run DIRECGEN. If DIRECGEN is rerun to add the directory requirements of additional products, you might have to make manual updates to the CP directory.*

6. *You cannot abbreviate control statements in the input CP directory file. You can abbreviate operands to these statements.*

Running DIRECGEN

To Invoke DIRECGEN

You can run DIRECGEN in panel mode, line mode, or by specifying products directly.

The format of the DIRECGEN command is:

DIRECGEN	[*prodspec1* [*prodspec2* ...*prodspecn*] [*(options* [*)*]]
	options: [**LIST** *fn ft* [*fm* / *****]] [**FILE** [*fn* / **VMUSERS**] [*ft* / **DIRECT**] [*fm* / *****]]

prodspec
: are product specification codes that let you specify the products for which you want DIRECGEN to generate directory resources. These codes consist of the product number and the feature identification code as listed in the FEATURE$ PRODUCTS file. If no feature identification code exists for a product, specify just the product number. If one exists, attach it to the end of the product number. Specify these codes without imbedded hyphens or other punctuation.

LIST
: tells DIRECGEN to get the products for which directory resources are to be allocated, from the file *fn ft fm*. If you do not specify a filemode, all accessed disks are searched. This file must be in the same format as the FEATURE$ PRODUCTS file.

FILE
: specifies the name of the CP directory updated by DIRECGEN. The default is VMUSERS DIRECT *.

Using the DIRECGEN Panels

- Invoke DIRECGEN from a full-screen terminal without product specification codes or the LIST option.
- When a panel listing optional feature program products appears on your screen, type an X next to the products you want to represent in the CP directory.

Note: See "DIRECGEN Panels" on page 10-28.

Using DIRECGEN on a Line Mode Terminal

- Copy FEATURE$ PRODUCTS into a new file.
- Edit the new file by deleting each line that lists a product you do not want to represent in the CP directory, and file the edited version.
- Invoke DIRECGEN using the LIST option to specify the name of the file you just created and edited.

Specifying Products Directly

You can specify product specification codes as arguments when you invoke DIRECGEN instead of listing them in a file or selecting them from a panel.

To do this, list the product specification codes as you find them in the FEATURE$ PRODUCTS file. Omit hyphens or other punctuation marks, and leave one blank after DIRECGEN and between each code. You can specify up to 130 characters, including the command and options, on the CMS command line.

After Running DIRECGEN

After you run DIRECGEN, do the following:

- If you received message 835W, refer to the DIRECGEN HISTORY file for additional updates that you must make to the CP directory.
- Type in DISKMAP VMUSERS DIRECT to execute the DISKMAP EXEC (optional). DISKMAP summarizes the MDISK statements in the CP directory file. Review the DISKMAP results before you continue. DISKMAP also identifies gaps and overlaps between the MDISK statements. Type HELP DISKMAP for more information.
- If you find, by reviewing the DIRECGEN HISTORY file, that you must increase the size of an existing minidisk and you do not have DIRMAINT, do the following:

  1. Find space for a minidisk the size of the original plus the needed expansion size, using DISKMAP output.
  2. Define a new minidisk in the CP directory using a temporary virtual address.
  3. Run DISKMAP again to verify.
  4. Place the CP directory on-line by issuing DIRECT *fn ft*
  5. Link and format the new minidisk.
  6. Copy the contents of the original minidisk to the new minidisk.

7. Verify the contents of the new minidisk.

8. CP format the original minidisk extents, and delete the minidisk from the CP directory.

9. Change the address of the new minidisk to the address of the original.

10. Run DISKMAP again to verify.

11. Place the CP directory on-line by issuing DIRECT *fn ft*

12. Detach and relink the new minidisk.

- You can tailor the directory before installation for better use of DASD space. For example, you can move minidisks you plan to use more often than others to the center of a DASD volume.

  Use the DISKMAP EXEC to produce a directory map before putting the directory on-line. Once you make all your changes, process the directory file with the following CMS command to see if it follows the required directory format:

  ```
  DIRECT VMUSERS DIRECT (EDIT
  ```

Notes:

1. *If DIRMAINT is installed, you can use the DIRMAINT CMDISK command to increase the size of an existing minidisk.*

2. *You can run DIRECGEN more than once, but you should select all products you plan to install the first time. If you run DIRECGEN more than once, you might have to manually update the directory.*

3. *DIRECGEN does not expand existing minidisks in the input CP directory file.*

4. *DIRECGEN does not add non-MDISK control statements to a userid if it conflicts with an existing statement. Such control statements are recorded in the DIRECGEN HISTORY file.*

5. *DIRECGEN sorts the MDISK statements for each userid in ascending EBCDIC order by virtual address. If necessary, do additional manual sorting.*

DIRECGEN HISTORY File

The DIRECGEN HISTORY file is updated each time you run DIRECGEN. This file lists products with directory resources defined by DIRECGEN and contains other information such as time stamps, error messages, and return codes. It also contains information about manual directory updates you might have to make after you run DIRECGEN.

The following is an example of a DIRECGEN HISTORY file after DIRECGEN is run twice:

```
--------------------------------------------------------------------------
Time and Date of Entry:  18:33:19     5 AUG 1986

Return Code from DIRECGEN EXEC:  0

Complete directory entries have been made for the following products:

   5668899       APL2
   5748XE4       DIRMAINT
   5748XXB       DMS/CMS
   5664283       VM/IS Productivity Facility
--------------------------------------------------------------------------
Time and Date of Entry:  18:37:28     26 AUG 1986

Return Code from DIRECGEN EXEC:  4

Complete directory entries have been made for the following products:

   5664296       CVIEW
   5664282       ISPF

The directory entries for the following products ARE NOT complete:

   5664200       GDDM/VM

   ********************************************************************
   *            PLEASE NOTE, VERY IMPORTANT!!
   *
   *  The following additions MUST BE made to the CP directory
   *  file to complete the directory entries for these products:
   *
   *     -ADD 9020 1K BLOCKS TO MAINT'S 19E
   ********************************************************************
```

Backing Out and Restarting DIRECGEN

If you have updated the CP directory since the last time you ran DIRECGEN, do not try to back out and restart DIRECGEN or you will lose these updates. In addition, optional feature program products might be erased when you restart DIRECGEN and place a new directory on-line.

If you want to undo directory updates made during the last successful execution of DIRECGEN, do the following:

1. Logon to MAINT userid.

2. Make sure VMOLD DIRECT exists.

Note: During the last execution of DIRECGEN, the input CP directory was saved as VMOLD DIRECT.

3. Erase VMUSERS DIRECT.

4. Rename VMOLD DIRECT to VMUSERS DIRECT.

   Note: If the name of your CP directory is not VMUSERS DIRECT, substitute the name of your directory.

5. XEDIT DIRECGEN $SELECT$ file and delete the entries for products selected in the last successful execution of DIRECGEN.

6. Invoke DIRECGEN.

7. If optional feature program products have not been installed and new minidisks have not been formatted since the last time you ran DIRECGEN, continue the steps in "After Running DIRECGEN" on page 8-11.

   If optional feature program products have been installed or new minidisks have been formatted since the last time you ran DIRECGEN:

   a. Enter DIRECT VMUSERS to place the directory on-line.

   b. Logoff.

   c. Logon to MAINT userid.

   d. Compare the new VMUSERS DIRECT with VMOLD DIRECT, and format new minidisks before beginning product installation.

   e. Continue the steps in "After Running DIRECGEN" on page 8-11.

Using the PASSMOD EXEC to Change Logon Passwords

PASSMOD changes logon passwords for userids in the CP directory file.

Invoke PASSMOD after you execute DIRECGEN and before you install optional feature program products with INSTFPP.

Before Running PASSMOD

Before you run PASSMOD, make sure:

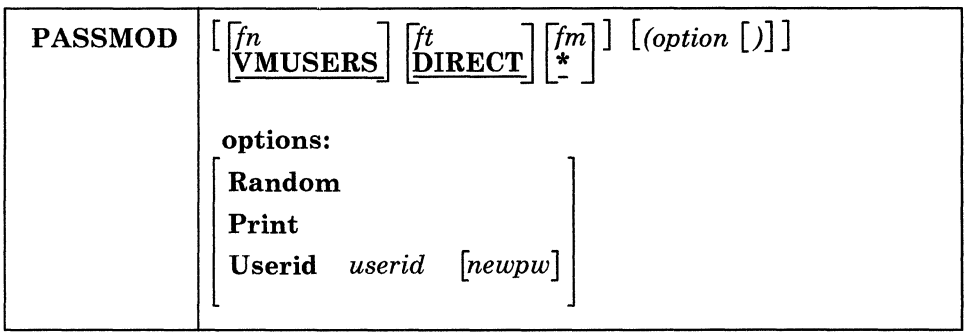
- DIRMAINT is not operational
- You have read/write access to the CP directory file
- The CP directory file has fixed length, 80-character records
- USER control statements in the input CP directory file are uppercased and are not abbreviated.

Running PASSMOD

To Invoke PASSMOD

You can run PASSMOD in panel mode, line mode, or by entering userids on the command line.

The format of the PASSMOD command is:



fn
is the filename of the CP directory source file. The default is VMUSERS.

ft
is the filetype of the CP directory source file. The default is DIRECT.

fm
is the filemode of the CP directory source file. You need read/write access to the disk containing the file. The default is *.

Random
specifies that passwords are generated at random for userids in the CP directory file that have the password NOLOG and do not begin and end with '$'. When the passwords have been generated, the panel is displayed with the current passwords and the new random passwords. Then, you can make more changes to the passwords. If your terminal is in line mode, you can change new random passwords individually before execution.

Print
specifies that the list of userids and passwords is printed in the order of the userids in the CP directory. The panel is not displayed, and line mode is not entered.

PASSMOD produces virtual printer spool files with the PRINT command. Make sure appropriate spooling control options are in effect to direct output to the desired real printer. You might have to invoke the CP SPOOL command and/or the CP TAG command.

Userid *userid*
specifies the name of a userid with a current password you want to change. You are prompted for the new password. The new password is written to the CP directory file.

newpw
specifies a new password that is written to the CP directory file.

Note: You cannot change the logon passwords of userids that begin and end with '$'.

Using the PASSMOD Panel

- Invoke PASSMOD from a full-screen terminal without specifying the USERID or PRINT options.

- When a panel displays the product IDs with their userids and current logon passwords, change the passwords according to the explanation on the panel.

  Note: You can enter passwords in panel mode with imbedded periods, but you cannot enter passwords with leading or trailing periods. To enter passwords with leading or trailing periods, invoke PASSMOD with the USERID option.

Using PASSMOD on a Line Mode Terminal

- Invoke PASSMOD.
- As you are prompted, enter a new logon password for each userid and corresponding password.
- To exit PASSMOD, enter either EXECUTE or QUIT, but do not abbreviate these.

  If you enter EXECUTE, new passwords are written to the CP directory file. Invoke the CMS DIRECT command to place the changes on-line.

  If you enter QUIT, PASSMOD prompts you to enter QUIT again to confirm that you do not want to save the passwords. If you enter QUIT again, PASSMOD exits, leaving the CP directory file unchanged.

Note: If you want to change a logon password to QUIT or EXECUTE, invoke PASSMOD with the USERID option.

Entering Userids on the Command Line

To change only one password, invoke PASSMOD with the USERID option and enter the userid and new password or just the userid on the command line. If you do not specify a password, a prompt is displayed to let you enter a new password. The new password is verified and placed in the CP directory file. Use the CMS DIRECT command to place the directory on-line for the password change to take effect.

After Running PASSMOD

After you run PASSMOD, do the following:

- Type in DIRECT VMUSERS to place directory on-line.
- Logoff.
- Logon to MAINT userid.
- Use the following example to format common minidisks 19E, 326, and 348 if they are present in your directory and have not already been formatted:

format 19e c Mode is variable.

```
DMSFOR603R FORMAT will erase all files
on disk C(19E). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
```

1

```
DMSFOR605R Enter disk label:
```

mnt19e

```
Formatting disk C
20 cylinders formatted on C(19E)
Ready;
```

Using the INSTFPP EXEC to Install Optional Feature Program Products

Note: Before using INSTFPP to install optional feature program products, read "Considerations for Installing Optional Feature Program Products" on page 8-25.

Before Running INSTFPP

Before you invoke INSTFPP to install optional feature program products, make sure:

- You are logged on to MAINT userid
- MAINT has all privilege classes
- MAINT has the MAINT 319 minidisk linked as read/write
- MAINT 190, MAINT 191, MAINT 193, and MAINT 194 are linked and the INSTFPP code is on one of these
- You mount the optional feature product tape. INSTFPP stops if the tape is not mounted correctly.
- You have a hardcopy of your directory available. Many product installation execs link to the user minidisks in write mode. If the link attempt fails, you might be asked to enter the write password of the minidisk.
- Enter INSTFPP (NOINSTALL MEMO to print the product Memo to Users for reference and to copy the Memo to Users to the MAINT 319 minidisk
- Make sure appropriate spooling control options are in effect to direct the virtual printer spool files INSTFPP produces with the PRINT command to the desired real printer. You might have to invoke the CP SPOOL command and/or the CP TAG command.

  If your printer handles only uppercase characters, use the FOLD option of the CP LOADBUF command. If your printer does not accept the LOADBUF command, print memos by issuing the PRINT command with the UPCASE option. In addition, if your printer cannot print special characters contained in the product memos printed by INSTFPP, look on-line at the product Memo to Users on the MAINT 319 minidisk.

  Refer to the *VM/SP CP Command Reference* and the *VM/SP CMS Command Reference* for more information about these commands.
- Your virtual storage size is 16M unless otherwise specified by the product Memo to Users.

Running INSTFPP

To Invoke INSTFPP

You can run INSTFPP in panel mode or by specifying products directly.

The format of the INSTFPP command is:

INSTFPP	[*prodspec1* [*prodspec2* ...*prodspecn*]] [*(options* [*)*]]
	options:
	[**Prompt** / **NOPrompt**] [**Memo** / **NOMemo**]
	[**Install** / **NOInstall**] [**Rewind** / **NORewind**]
	[**All**] [*rdev*]

prodspec
are the product specification codes that let you specify the products you want processed. These codes consist of the product number and the feature identification code as listed in the FEATURE$ PRODUCTS file. If no feature identification code exists for a product, specify just the product number. If one exists, attach it to the end of the product number. Specify these codes without imbedded hyphens or other punctuation. INSTFPP scans the files on the stacked tape and processes the selected optional feature program products.

Prompt
displays the prompts that ask if you want to process the specified optional feature program products. Prompt is the default.

NOPrompt
eliminates the prompts that asks if you want to install the specified optional feature program products.

Memo
prints a product Memo to Users from the tape for each selected product. Memo is the default.

NOMemo
lets you process the selected products without printing product Memo to Users. You cannot specify NOMemo if you have specified NOInstall.

Install
lets you install the selected products. Install is the default.

NOInstall
lets you process the selected products without installing them. You cannot specify NOInstall if you have specified NOMemo.

Rewind
makes sure the tape is rewound before and after product installation and that it is properly mounted. Rewind is the default.

NORewind
lets INSTFPP processing continue without tape rewinds before and after product installation. You can specify NORewind only if you invoke INSTFPP from the command line.

Note: Make sure the tape is properly mounted and attached as 181. You can only install products located after the initial tape position.

All
lets you process all the products on the tape. All is the default if you do not enter product specification codes.

rdev
lets you specify the real tape address if the tape is mounted on a tape drive but not attached to MAINT as 181.

Using the INSTFPP Panels

- Invoke INSTFPP with no arguments from a 3270 device (20 line minimum).
- When a panel appears on your screen, enter the real tape drive address, change defaults if necessary, and press ENTER.
- If you choose not to install all products on the product tape, type an X next to the products you want to install.

Note: See "INSTFPP Panels" on page 10-32.

Specifying Products Directly

If you enter arguments, the INSTFPP panels do not appear on your screen. You can specify product specification codes by listing them as you find them in the FEATURE$ PRODUCTS file. Omit hyphens or other punctuation marks, and leave one blank between each code. You can specify up to 130 characters, including the command and options, on the CMS command line.

After Running INSTFPP

After you run INSTFPP, do the following:

- Execute manual installation and verification procedures as indicated in the product Memo to Users if necessary.
- Tailor product dependent files as indicated in the product Memo to Users if necessary.
- Resave CMS if optional feature program products that you installed loaded files to the MAINT 19E minidisk.
- Consider placing file directory information for shared, read/only minidisks into a DCSS using the SAVEFD command. Refer to *VM/SP CMS for System Programming.*

Notes:

1. *INSTFPP cannot properly restore minidisks accessed as read/only extensions with a subset defined. INSTFPP reaccesses minidisks as read/only extensions with no subset specification.*
2. *INSTFPP leaves the tape drive containing the optional feature product tape attached as virtual address 181.*
3. *A console file is created during the installation process to record terminal activity and is spooled to the MAINT reader when INSTFPP is complete.*
4. *At least 30 contiguous cylinders of 3380 temporary disk space or equivalent must be available.*

PROD LEVEL File

INSTFPP updates a file named "PROD LEVEL" on the MAINT 319 minidisk with the results of each optional feature program product installation.

The following is an example of what a PROD LEVEL file can look like:

```
--------------------------------------------------------------------------
5748XXB  Display Management System (DMS/CMS)
VER 1    REL 2     MOD 0
Time and date of entry: 20:25:25   30 Sep 1986
*** Product installed and verified successfully
--------------------------------------------------------------------------
5799AXX  3277 Graphics Attachment Support PRPQ (GASP)
VER 1    REL 3     MOD 0
Time and date of entry: 21:17:25   30 Sep 1986
*** Product installed; manual verification required
```

Update Messages

Each optional feature program product entry in the PROD LEVEL file has an update message associated with it. The possible update messages and their explanations are:

***** Product installed and verified successfully**
The optional feature program product installed correctly, and the product was verified successfully.

***** Product files loaded; see the Memo to Users to complete installation**
The optional feature program product files have been loaded successfully. Refer to the product Memo to Users printed by INSTFPP. This memo tells you how to complete the installation of the product and then verify that it installed correctly. In some cases, the product Memo to Users refers to other documentation.

***** Product installed; manual verification required**
The optional feature program product installed, but it was not verified automatically. Refer to the product Memo to Users printed by INSTFPP. This memo tells you how to make sure the optional feature program product installed correctly. In some cases, the product Memo to Users refers to other documentation.

***** Product installed; verification failed**
The optional feature program product installed, but the automatic verification failed. Try to install the optional feature program product again after correcting any problems; if it does not verify correctly, contact your support personnel.

***** Product Installation EXEC failed; RC =** *rc*
The product installation exec failed, and the return code passed back by this exec to INSTFPP is *rc*. Refer to the product Memo to Users or product installation exec prologue to see what this return code means. If you cannot fix the problem, contact your support personnel.

Rerunning INSTFPP

If any optional feature program products do not install correctly, do the following:

- Try to solve the problem by using the console log, product Memo to Users, the PROD LEVEL file, and other product specific documentation.
- Ready the tape.
- Invoke INSTFPP.
- Reinstall products that did not install correctly. (Also reinstall products that have these products as prerequisites. Refer to "Optional Feature Program Products" on page 9-1 to see if products you are reinstalling are prerequisites for other products you are installing.)

After you install each optional feature program product, follow the instructions in the product Memo to Users to verify that it has installed correctly (unless it was automatically verified during installation.)

Considerations for Installing Optional Feature Program Products

Notes:

1. *If optional feature program products you want to install load files to MAINT's 19E minidisk, resave CMS when the installation is complete.*

2. *Do not install optional feature program products in full-screen CMS, or errors might occur.*

3. *When you install optional feature program products, your system national language should be American English.*

Review these considerations before you install optional feature program products:

ACF/NCP (5668-854)

During the installation process, you enter an XEDIT session to update the 5668854 VMFPARM file. If you are using the installation defaults, enter QQUIT to exit from this XEDIT session.

If you are also installing EP (5735-XXB), the disk addresses in EP's 5735XXBC VMFPARM file must be the same as those in NCP's 5668854 VMFPARM file. Also, the 5668854A VMFPARM file for common code must have the same disk addresses as defined in 5668854 VMFPARM.

ACF/SSP (5664-289)

If you plan to use the Configuration Control Program (CCP) facility of SSP, install ISPF (5664-282) before SSP.

During the installation process, you enter an XEDIT session to update the 5664289A VMFPARM file. If you are using the installation defaults, enter QQUIT to exit from this XEDIT session.

ACF/VTAM (5664-280)

Install the Group Control System (GCS) before VTAM. In addition, when you define your GCS group configuration file:

- Make VTAM an authorized userid for GCS
- Define the VTAM Discontiguous Saved Segment (DCSS) as a shared segment linked automatically when you IPL your GCS system.

During the installation process, you enter an XEDIT session to update the 5664280A VMFPARM file. If you are using the installation defaults, enter QQUIT to exit from this XEDIT session.

ACRITH (5664-185)

Install ISPF (5664-282) and VS FORTRAN (5668-806) before ACRITH.

APL2 (5668-899)

APL2 provides optional support for SQL/DS (5748-XXJ), GDDM/VM (5664-200), and GDDM-PGF (5668-812). If you have any of these products, install them before APL2 to enable this support.

You must execute the manual verification for APL2 from a userid other than MAINT with a virtual storage size of 2M.

AS (5767-032)

Install GDDM/VM (5664-200) and GDDM-PGF (5668-812) before you install AS. If using SQL/DS (5748-XXJ) or ISPF (5664-282), install these before AS.

CSP/AD (5668-813)

Install CSP/AE (5668-814) and VSE/VSAM (5746-AM2) before CSP/AD. If the second part of the installation has not been successfully completed, restart the installation process.

CSP/AE (5668-814)

Install VSE/VSAM (5746-AM2) before CSP/AE. If the second part of the installation has not been successfully completed, restart the installation process.

CSP/Q (5668-918)

Install CSP/AD (5668-813), CSP/AE (5668-814), and VSE/VSAM (5746-AM2) before CSP/Q.

CU2 (5798-RWL)

Install GDDM/VM (5664-200) and PL/I Compiler and Libraries (5734-PL3) before CU2. If you install PL/I Compiler and Libraries, install both the transient and resident libraries in PLILIB TXTLIB.

You cannot run the verification procedure from the operator's console.

DIRMAINT (5748-XE4)

After installation, edit the DVHDMX XEDIT file to insert your DATAMOVE Virtual Machine password for the operand &PASS=password. At installation, the DVHDMX XEDIT file is loaded to DIRMAINT's 191 minidisk.

DisplayWrite/370 (5664-370)

Sufficient DASD space has been allocated to install DisplayWrite/370 with three national languages besides American English. If you want to install additional lanuages,

refer to the product Memo to Users for additional space requirements.

EP (5735-XXB)

During the installation process, you enter an XEDIT session to update the 5735XXBC VMFPARM file. If you are using the installation defaults, enter QQUIT to exit from this XEDIT session.

If you are also installing NCP (5668-854), the disk addresses in NCP's 5668854 VMFPARM file must be the same as those in EP's 5735XXBC VMFPARM file. Also, 5668854A VMFPARM file for common code must have the same disk addresses as those defined in 5735XXBC VMFPARM.

GDDM/IMD (5668-801)
GDDM/PGF (5668-812)
GDDM/VM (5664-200)

If you are installing APL2 (5668-899), install the GDDM products before APL2 to include the GDDM support in APL2.

Install GDDM/VM (5664-200) before GDDM-PGF (5668-812) and GDDM-IMD (5668-801).

Do not install the GDDM/VM NL tape until you install the GDDM/VM tape.

Do not install the GDDM-PGF NL tape until you install the GDDM-PGF tape.

When you install the GDDM/VM NL and GDDM-PGF NL tapes, you are prompted for the national languages you want to load from the tape. The MAINT 19E disk and predefined DMKSNT entries include enough space to support one language besides American English.

Generate the DCSSs for GDDM/VM, GDDM-PGF, and GDDM-IMD using the ADMBLSEG EXEC. Before you invoke ADMBLSEG, make sure you reIPL CMS and invoke the SET LDRTBLS 8 command, or the DCSS build can fail. The DMKSNT entries have enough space reserved for each product to include all optional parts in each DCSS.

IC/1 (5668-897)

Install OS PL/I transient library (5734-LM5), APL2 (5668-899), GDDM/VM (5664-200), GDDM/PGF (5668-812), and APE V2 (5668-808) before IC/1. Also, if you have SQL/DS (5748-XXJ), install it before IC/1. During the installation of IC/1, you are asked if you want to build the sample SQL tables for IC/1. If you answer YES, the D5668897 EXEC is invoked. This exec fails if the SQL disconnected machine, SQLDBA, is not logged on and properly initialized. You can reexecute this exec if it fails during

installation. Enough DASD space has been allocated to support one national language besides American English.

Throughout IC/1 installation, APL2 is entered. If you need to clear the screen during APL2 sessions, use the CLEAR key. Do not use the PA2 key, because this suppresses APL2 output to the screen. If an error occurs and a VM READ state is entered, enter the following command in uppercase:

```
)OFF HOLD
```

The installation failed, so take the necessary corrective steps.

ISPF (5664-282)

Sufficient DASD and DMKSNT space has been allocated to install ISPF with three national languages besides American English in MACLIB format only. If you want to install additional languages, or if you want to install CMS files rather than or in addition to MACLIBs, refer to *ISPF and ISPF/PDF Installation and Customization*, SC34-4015, for additional space requirements.

ISPF/PDF (5664-285)

Sufficient DASD space has been allocated to install ISPF/PDF with two languages besides American English in MACLIB format only. If you want to install CMS files rather than or in addition to MACLIBs, refer to *ISPF and ISPF/PDF Installation and Customization Manual*, SC34-4015, for additional space requirements.

NetView (5664-204)

Install VSE/VSAM (5746-AM2) before NetView.

Make NETVIEW an authorized userid for Group Control System (GCS) when you define your GCS group configuration file.

During the installation process, you enter an XEDIT session to update the 5664204 VMFPARM file. If you are using the installation defaults, enter QQUIT to exit from this XEDIT session.

The following considerations apply to the manual verification tasks outlined in the *NetView Installation and Administration Guide*, SC30-3360:

- CNMSV003 DSF and CNMSV004 EXEC must be modified unless you are using 3350 DASD.
- Make sure the MAINT and NETVIEW userids are never logged on at the same time. This ensures that all required read/write links are present when executing verification tasks.

- The CNMNDEF EXEC must be invoked with parameter BOTH=01.
- The DSIPRT EXEC must be invoked AFTER the operator tasks in Chapter 5 are performed.
- Invoke the CP ENABLE SNA command from the MAINT userid before starting VTAM.
- Make sure the GCS recovery machine (usually GCS) is logged on before you try to issue tasks requiring the GCS environment.
- Be patient after issuing the NETSTRT command. Command output appears after the ready message is displayed.
- Before invoking the VMVTAM command, invoke the CP DEFINE GRAF 441 3270 command from the VTAM userid.
- Before performing operator tests, invoke the DIAL VTAM 441 command from the VM logon screen to access the NetView operator terminal.

OGL (5664-293)
If you are installing PSF/VM (5664-198), install it before OGL.

OS PL/I Optimizing Compiler and Libraries (5734-PL3)
OS PL/I Transient Library (5734-LM5)
During installation you are prompted for the disk address and filemode used for the installation. Respond 19E and E respectively. Make sure the MAINT 19E disk is not accessed as a read/only extension before invoking INSTFPP, or the installation exec will fail.

PC BOND (5664-298)
Install VM Bond on personal computers that use PC Bond.

PROFS (5664-309)
Make sure the volume PROFPK has been formatted and allocated before you install PROFS. Refer to "Formatting and Allocating DASD Volumes" on page 8-4 for more information.

PSAF (5664-312)
Install PSF/VM (5664-198) before you install PSAF.

PSF (5664-198)
Verification cannot be done until you install the basic feature and either the 3800 or 3820 features.

QMF (5668-AAA)
Work with your installation's Data Base Administrator or someone familiar with SQL/DS to install QMF. The QMF installation process supplies default values to simplify the

installation, but refer to the *QMF: Installation Guide*, SC26-4243, to estimate your data base requirements.

RSCS V2 (5664-188)

Generate Group Control System (GCS) before you install RSCS V2. Make RSCSV2 an authorized user in the GCS group when you define your GCS group configuration file. The GCS recovery machine must be running before the RSCS V2 machine. Refer to the *Remote Spooling Communications Subsystem Networking Version 2 Planning and Installation Guide*, SH24-5057.

MAINT 300 and MAINT 310 are created when DIRECGEN is run for RSCS V2 or IPF. These disks hold the files needed for IPF to interface with RSCS. If you are not installing IPF, you can respond PASS to the prompt in the RSCS V2 installation that asks if the RSCS IPF files should be loaded. After you install RSCS V2, you can delete the 300 and 310 disks if IPF is not installed.

MAINT 39F is also created when you run DIRECGEN for RSCS V2. This disk contains the RSCS V2 transmission algorithm source, which is shipped on the product tape, and has enough space to contain the optional RSCS V2 source that is on a separately ordered tape. After you install RSCS V2, you can reduce the size of this disk if you do not want to install the source. Or you can delete the disk if you do not want the transmission algorithm source files.

SQL/DS (5748-XXJ)

If you are installing APL2 (5668-899) and SQL/DS, install SQL/DS before APL2 if you want APL2 to support SQL/DS.

If you are using the SQLPK volume to install SQL/DS, make sure the volume has been formatted and allocated. Refer to "Formatting and Allocating DASD Volumes" on page 8-4 for more information.

The service level of SQL/DS includes remote relational access support. If you want to use this facility, the CP directory entry for the SQL/DS service machine might have to be updated. Refer to the *SQL/Data System Remote Relational Access Support Guide for VM/System Product (SHO9-8016)*, and other SQL/DS books for more information.

VMBACKUP-MS (5664-291)

If you order this product through the System Offering, do not follow the installation process outlined in the *VMBACKUP Management System Installation Guide (SH20-6249)*. Instead, follow the installation process outlined in the product Memo to Users on the product tape.

VMMAP (5664-191)

Install OS PL/I Transient Library (5734-LM5) or OS PL/I Optimizing Compiler and Libraries (5734-PL3) before you install VMMAP. Install the transient library in PLILIB TXTLIB.

VM/RTM (5796-PNA)

You cannot run the verification procedure for the userid SMART from the operator's console.

VMTAPE-MS (5664-292)

If you order this product through the System Offering, do not follow the installation process outlined in the *VMTAPE Management System Installation Guide (SH20-6241).* Follow the installation process outlined in the product Memo to Users on the product tape.

VM3812 (5798-DTE)

If you have ordered DCF (5748-XX9), install it before you install VM3812. Refer to the *VM3812- IBM 3812 Pageprinter VM Support Installation Guide,* SH20-6750, for DCF considerations.

If you have ordered VM/IS Productivity Facility (5664-283) and want to include VM3812 support in the panels, do not run the panel generation exec, ESSSETUP, until you have installed both products. This process can take some time to complete. Also, you must make sure CMSUSER has a link to the VM/IS Productivity Facility product disk before you can run ESSSETUP.

VS FORTRAN (5668-806)

If you want to install a copy of the composite module AFBVRENC in a Discontiguous Saved Segment (DCSS), use the FTNLIB10 DCSS supplied in the sample DMKSNT file. You are prompted for the name of this segment during installation.

During the execution of the verification exec, V5668806 EXEC, you are placed in an interactive debug session. Test desired debug commands, and enter QUIT to exit from this session.

If you want to use VSAM I/O, install VSE/VSAM (5746-AM2).

VSE/VSAM (5746-AM2)

Before you install VSE/VSAM, generate the CMSDOS and CMSBAM shared segments. Before you invoke INSTFPP, MAINT's virtual storage must be defined at least 512K above the CMSAMS and CMSVSAM Discontiguous Saved Segments (DCSS) and below the CMSDOS DCSS. If the sample DMKSNT entries are being used, define MAINT's virtual storage to 15M.

Chapter 9. Information About Optional Feature Program Products

Optional Feature Program Products

VM/SP System Offering 5.0 supports the following list of optional feature program products. The list shows the abbreviations, numbers, and release levels of each product. Also shown are prerequisites, products needed to install or operate the listed products.

Product	Abbreviation	Number	Level	Prerequisite
ACF/Network Control Program	ACF/NCP	5668-854	4.2.0	5664-280 5664-289
ACF/System Support Program	ACF/SSP	5664-289	3.2.0	5664-282
ACF/Virtual Telecommunications Access Method	ACF/VTAM	5664-280	3.1.1	GCS[1]
APL2		5668-899	1.2.0	
Application Prototype Environment	APE V2	5668-808	2.1.0	5664-200 5668-812 5668-899
Application System	AS	5767-032	1.4.0	5664-200 5668-812
Composition Utility Version 2	CU2	5798-RWL	1.1.0	5664-200 5734-PL3
Contextual File Search/370	CFSearch/370	5664-329	1.1.1	
Cooperative Viewing Facility Version 2	CVIEW	5664-296	2.1.1	
Cross System Product/Application Development	CSP/AD	5668-813	3.1.0	5746-AM2 5747-DS1[2] 5668-814
Cross System Product/Application Execution	CSP/AE	5668-814	3.1.0	5746-AM2 5747-DS1[2]

Product	Abbreviation	Number	Level	Prerequisite
Cross System Product/Query	CSP/Q	5668-918	1.2.1	5746-AM2 5747-DS1[2] 5668-813 5668-814
Display Management System for CMS	DMS/CMS	5748-XXB	1.2.0	
Display Write/370	DisplayWrite/370	5664-370	1.1.1	
Document Composition Facility	DCF	5748-XX9	1.3.0	
Emulation Program	EP	5735-XXB	1.4.0	5664-289
Environmental Recording Editing and Printing Program	EREP	5654-260	3.3.0	
Font Library Service Facility	FLSF	5668-890	1.1.0	
FORTRAN Utilities		5798-DFH	1.2.2	5668-806
Graphical Display and Query Facility	GDQF	5668-905	1.4.0	5664-200 5664-282 5664-285
Graphics Attachment Support Program	GASP	5799-AXX	1.3.1	5668-806
Graphic Data Display Manager Base	GDDM/VM	5664-200	2.1.0	
Graphic Data Display Manager/VM National Language Feature	GDDM/VM NL	5664-200	2.1.0	5664-200 (GDDM/VM)
Graphic Data Display Manager/ Interactive Map Definition	GDDM-IMD	5668-801	2.1.0	5664-200
Graphic Data Display Manager/ Presentation Graphics Function	GDDM-PGF	5668-812	2.1.0	5664-200
Graphic Data Display Manager/ Presentation Graphics Function National Language Feature	GDDM-PGF NL	5668-812	2.1.0	5664-200 5668-812 (GDDM-PGF)
Host-Displaywriter Document Interchange	HDDI	5799-BKE	1.2.0	5734-LM5 or 5734-PL3 5748-XX9
IBM BASIC/VM		5668-996	1.2.0	
IBM High-Accuracy Arithmetic Subroutine Library	ACRITH	5664-185	1.3.0	5664-282 5668-806
IBM 3812 Pageprinter VM Support	VM3812	5798-DTE	1.1.1	

Product	Abbreviation	Number	Level	Prerequisite
Info Center/1	IC/1	5668-897	1.1.3	5664-200 5668-812 5668-899 5668-808 5734-LM5 or 5734-PL3
Interactive Instructional Presentation System	IIPS	5668-012	1.2.0	
Interactive System Productivity Facility	ISPF	5664-282	2.2.0	
ISPF/Program Development Facility	ISPF/PDF	5664-285	2.2.0	5664-282
NetView		5664-204	1.1.0	5664-280 5746-AM2
OS PL/I Optimizing Compiler and Libraries	PL/I Comp and Lib	5734-PL3	1.5.1	
OS PL/I Transient Library	PL/I Transient Lib	5734-LM5	1.5.1	
Overlay Generation Language	OGL	5664-293	1.1.0	
Page Printer Formatting Aid	PPFA	5664-199	1.1.0	
Pascal/VS		5796-PNQ	1.2.2	
Personal Computer Bond[3]	PC BOND	5664-298	1.2.1	
Print Services Access Facilities	PSAF	5664-312	1.1.0	5664-198 5664-282
Print Services Facility/VM Base	PSF/VM Base	5664-198	1.1.0	
Print Services Facility/VM 3800	PSF/VM 3800	5664-198	1.1.0	5664-198 (PSF/VM Base)
Print Services Facility/VM 3820	PSF/VM 3820	5664-198	1.1.0	5664-198 (PSF/VM Base) 5664-280 5664-289 5668-854
Professional Office System	PROFS	5664-309	2.2.1	
Professional Office System Applications Support Feature	PROFS ASF	5664-309	1.1.0	5664-282 5664-309 (PROFS)
Query Management Facility	QMF	5668-AAA	2.1.0	5664-282 5748-XXJ 5664-200 5668-812

Product	Abbreviation	Number	Level	Prerequisite
Remote Spooling Communications Subsystem Networking, Version 1	RSCS (version 1)	5748-XP1	1.3.0	
Remote Spooling Communications Subsystem Networking, Version 2	RSCS (version 2)	5664-188	2.2.0	GCS[1]
Structured Query Language/Data System	SQL/DS	5748-XXJ	1.3.5	
Virtual Storage Extension/Virtual Storage Access Method	VSE/VSAM	5746-AM2	1.3.0	
VMBACKUP Management System	VMBACKUP-MS	5664-291	1.4.1	
VM Batch Facility	VMBATCH	5664-364	1.1.0	
VM Directory Maintenance	DIRMAINT	5748-XE4	1.2.0	
VM File Storage Facility	VMFSF	5798-DMY	1.1.3	
VM/Integrated System Productivity Facility	VM/IS-Productivity Facility	5664-283	1.5.0	5664-282 5664-318
VM/Interactive Productivity Facility	VM/IPF	5664-318	2.2.0	
VM Monitor Analysis Program	VMMAP	5664-191	1.1.3	5734-LM5 or 5734-PL3
VM/Pass-Through Facility	PVM	5748-RC1	1.3.0	
VM/Personal Computer Host Server[3]	VM/PC Host Server	5664-319	1.1.0	
VMTAPE Management System	VMTAPE-MS	5664-292	1.3.1	
VM/370 Real Time Monitor	VM/RTM	5796-PNA	1.1.7	
VS COBOL II	COBOL 2	5668-958	1.2.0	
VS FORTRAN	VSFORTRAN	5668-806	2.1.1	
3270 PC File Transfer[3]		5664-281	1.1.0	

[1] Group Control System (GCS) is provided with the VM/SP base.

[2] Device Support Facilities (5747-DS1) is provided with the VM/SP base.

[3] This product has PC software prerequisites. Refer to individual product documentation for prerequisite information.

Reference Books

Reference books are available for all products listed in "Optional Feature Program Products" on page 9-1. Refer to the VM/SP 5 ordering instructions for a list of available optional feature program product publications. Your IBM representative or nearest IBM branch office can help you obtain these books and can tell you how to subscribe to them so you automatically get updates and new editions.

Product Directory Information

The following table lists each optional feature program product, the userids and minidisks associated with each product, and the DASD space to be allocated for that minidisk. When you run DIRECGEN, the required space is allocated for the products you want to install.

Use the information in the following table to determine what userids, minidisks, and minidisk sizes are associated with the products you want to install.

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
ACRITH	MAINT	19E	9900
APE	MAINT	19E	10700
APL2	APL2PP	191	20000
	AP2SVP	191	1000
	MAINT	19E	7000
AS	VMASMON	191	200 blksz = 4K
	VMASSYS	191	2000 blksz = 4K
	VMASSYS	391	11000 blksz = 4K
	VMASSYS	392	500 blksz = 4K
	VMASSYS	393	3000 blksz = 4K
	VMASTEST	191	2000 blksz = 4K
CFSearch/370	MAINT	360	18600
COBOL 2	MAINT	19E	5382
CSP/AD	CSPUSER	191	5600
	CSPUSER	193	500 blksz = 4K
CSP/AE	CSPUSER	191	4000
	CSPUSER	193	200 blksz = 4K
	CSPUSER	502	9405
	CSPUSER	503	9405
CSP/Q	CSPUSER	191	5000
	CSPUSER	193	200 blksz = 4K
CU2	MAINT	19E	1750
CVIEW	CVIEW	191	450 blksz = 4K
DCF	MAINT	19E	3500
DIRMAINT	DATAMOVE	191	1100
	DIRMAINT	191	1800
	DIRMAINT	193	3876
	DIRMAINT	195	3900
	MAINT	19E	700
DisplayWrite/370	MAINT	361	12500
DMS/CMS	MAINT	19E	760

Figure 9-1 (Part 1 of 6). Product Directory Information

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
EP	MAINT	352	6672
	MAINT	353	990
	MAINT	354	1980
	MAINT	355	11250
	MAINT	356	990
	MAINT	348	75
EREP	MAINT	201	10260
FLSF	MAINT	363	2200
FORTRAN UTILITIES	MAINT	19E	147
GASP	MAINT	19E	601
GDDM-IMD	MAINT	19E	2038
GDDM-PGF	MAINT	19E	2366
GDDM-PGF NL	MAINT	19E	1800
GDDM/VM	MAINT	19E	8200
GDDM/VM NL	MAINT	19E	200
GDQF	MAINT	34A	6000 blksz = 4K
HDDI	MAINT	19E	1425
IBM BASIC	MAINT	19E	4400
IC/1	MAINT	19E	13050
IIPS	IIPS	191	5700
	IIPS	193	8700
ISPF	ISPVM	191	2000
	ISPVM	192	6975
ISPF/PDF	ISPVM	192	18135
NCP	MAINT	352	23653
	MAINT	353	3510
	MAINT	354	7020
	MAINT	355	29250
	MAINT	356	3510
	MAINT	348	150

Figure 9-1 (Part 2 of 6). Product Directory Information

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
NetView	MAINT	19E	6300
	MAINT	298	1800
	MAINT	330	44100
	MAINT	331	4500
	MAINT	332	9000
	MAINT	333	4500
	MAINT	334	44180
	MAINT	348	75
	NETVIEW	198	15750
OGL	MAINT	19E	310
	MAINT	363	1125
OS PL/I Optimizer Lib	MAINT	19E	8200
OS PL/I Transient Lib	MAINT	19E	600
Pascal/VS	MAINT	19E	1875
PC BOND	MAINT	19E	450
PPFA	MAINT	19E	100
	MAINT	363	400
PROFS	PROCAL	191	1440
	PROCAL	5FB	5760
	PROCAL	5FC	5760
	PROCAL	5FD	5760
	PROCAL	5FE	5760
	PROCAL	5FF	5760
	PRODBM	161	4800
	PRODBM	191	1440
	PRODBM	5FD	5760
	PRODBM	5FE	5760
	PRODBM	5FF	5760
	PROMAIL	151	1440
	PROMAIL	191	3360
	SYSADMIN	191	4800
	SYSADMIN	298	13440
	SYSADMIN	398	8640
	SYSADMIN	399	8640
PROFS ASF	SYSADMIN	397	900
	SYSADMIN	399	2250
PSAF	MAINT	19E	2200

Figure 9-1 (Part 3 of 6). Product Directory Information

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
PSF/VM - Basic Feature	MAINT	19E	48
	PSFMAINT	191	5115
	PSFMAINT	193	1860
	PSFMAINT	194	4650
	SFCM1	191	9300
PSF/VM - 3800 Feature	PDM470	191	2325
PSF/VM - 3820 Feature	PDMREM1	191	2325
PVM	MAINT	19E	150
	MAINT	29E	2800
	MAINT	36E	1400
	MAINT	39E	10500
	MAINT	49E	2100
QMF	MAINT	347	10479
	MAINT	346	5966
RSCS V1	MAINT	19E	100
	MAINT	39E	10000
	MAINT	49E	1000
	RSCS	191	768
RSCS V2	MAINT	19E	100
	MAINT	300	1024
	MAINT	310	1188
	MAINT	29F	512
	MAINT	39F	14080
	MAINT	49F	2304
	MAINT	59F	1280
SQL/DS	SQLDBA	191	4650
	SQLDBA	193	16275
	SQLDBA	195	1500 blksz = 4K
	SQLDBA	200	15810
	SQLDBA	201	1200 blksz = 4K
	SQLDBA	202	11550 blksz = 4K
	SQLUSER	191	1395
SSP	MAINT	33F	22108
	MAINT	340	4500
	MAINT	341	9000
	MAINT	342	4500
	MAINT	343	21390
	MAINT	348	120

Figure 9-1 (Part 4 of 6). Product Directory Information

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
VM/IPF	ADMIN	191	200
	CPRM	191	100
	CPRM	192	3000
	CPRM	291	340
	DISKACNT	191	168
	IPFSERV	191	465
	MAINT	19E	4135
	MAINT	300	4598
	MAINT	310	4604
	MAINT	3A0	418
	OP1	191	200
	SYSDUMP1	191	200
	VMUTIL	191	200
VM/IS-Productivity Facility	DEMO1	191	1395
	DEMO2	191	1395
	DEMO3	191	1395
	DEMO4	191	1395
	MAINT	19E	2500
	MAINT	322	3255
	MAINT	326	4650
VM/PC Host Server	MAINT	19E	300
VM/RTM	SMART	191	4600
VMBACKUP-MS	VMARCH	191	5000
	VMARCH	193	3000
	VMARCH	100	3000
	VMARCH	101	3000
	VMARCH	200	3000
	VMBACKUP	191	2500
	VMBACKUP	192	1000
	VMBACKUP	193	1000
	VMBACKUP	194	20000
	VMBSYSAD	191	2000
	VMBSYSAD	192	4000
VMBATCH	BATCH	191	540 blksz = 2K
	BATCH	193	4650 blksz = 2K
	BATCH	194	540 blksz = 2K
	BATCH	199	270 blksz = 2K
	BATCH	195	270 blksz = 2K
	BATCH1	191	1080 blksz = 2K
	BATCH2	191	1080 blksz = 2K
	MAINT	19E	700

Figure 9-1 (Part 5 of 6). Product Directory Information

Product	Product Userids	Minidisks	Size (CMS 1K blocks unless otherwise stated)
VMFSF	FSFADMIN	192	1000
	FSFCNTRL	191	3000
	FSFCNTRL	192	800
	FSFCNTRL	193	800
	FSFCNTRL	194	400
	FSFCNTRL	195	400
	FSFCNTRL	197	400
	FSFCNTRL	200	2000
	FSFCNTRL	201	2000
	FSFCNTRL	400	2000
	FSFCNTRL	401	2000
	MAINT	19E	270
VMMAP	VMMAP	191	10800
VMTAPE-MS	VMTAPE	191	2000
	VMTAPE	200	3000
	VMTAPE	300	3000
	VMTLIBR	191	2000
	VMTLIBR	192	3000
VM3812	VM3812	191	1860
	VM3812	192	750 blksz = 4K
	VM3812	193	2250 blksz = 4K
VSE/VSAM	MAINT	19E	950
VSFORTRAN	MAINT	19E	3250
VTAM	MAINT	19E	1700
	MAINT	298	2250
	MAINT	299	10800
	MAINT	29A	4500
	MAINT	29B	9000
	MAINT	29C	2250
	MAINT	29D	4500
	MAINT	348	75
3270 PC File Transfer	MAINT	19E	25

Figure 9-1 (Part 6 of 6). Product Directory Information

Userid Descriptions

This section gives a brief description of the userids associated with optional feature program products supported by System Offering.

- ADMIN (IPF)

  The ADMIN userid is reserved for the system administrator. The system administrator's tasks include authorizing users of the system and controlling the allocation of system resources (primarily DASD).

- APL2PP (APL2)

  APL2PP loads the APL2 product files, including the user common files copied to MAINT 19E.

- AP2SVP (APL2)

  The AP2SVP userid that runs as a disconnected virtual machine is the APL2 shared variable processor.

- AUTOLOG1

  VM/SP automatically logs on this virtual machine when the VM/SP system is IPLed. You can use the PROFILE EXEC for this userid to automatically do a set of actions every time the system is IPLed. These actions include automatically logging on other disconnected virtual machines, enabling lines and/or starting printers, and so on.

- CMSBATCH

  You can send time-consuming jobs to userid CMSBATCH, freeing your terminal for interactive work and keeping such jobs from slowing down the system. For more information about the CMSBATCH virtual machine, refer to the *VM/SP CMS User's Guide.*

- CMSUSER

  CMSUSER is an example of a virtual machine needed by typical CMS users. Create similar entries for each CMS user in your installation.

- CPRM (IPF)

  VM/IPF Problem Control Facility (PCF) needs the CPRM (Central Problem Report Manager) userid. This disconnected service machine controls and changes the problem data base.

CPRM minidisks are:

CPRM 191 used as a work disk.
CPRM 192 contains problem reports submitted and updated by users, problem reports created as the result of a dump, and control files.
CPRM 291 contains files needed to execute the VM/IPF Problem Control Facility dialogs.

- CSPUSER (CSP/AD, CSP/AE, CSP/Q)

  CSPUSER is an example of a userid needed by typical Cross System Product/Application Development (CSP/AD), Cross System Product/Application Execution (CSP/AE), and Cross System Product/Query (CSP/Q) users. Create similar entries for each CSP user in your installation.

- CVIEW (CVIEW)

  CVIEW Version 2 runs as a CMS application in a disconnected virtual machine, letting many users view a single interactive session. The CVIEW virtual machine gives identical screen images to session participants at the same time.

- DATAMOVE (DIRMAINT)

  The DATAMOVE virtual machine automates the process of copying user CMS minidisks and cleaning old minidisks before making them available for later allocation. DATAMOVE should be autologged with the DIRMAINT virtual machine at system IPL time.

- DEMO1 (VM/IS-Productivity Facility)

  The Project Engineer uses the DEMO1 userid in the system scenario described in *VM/IS Learning to Use Your System: A Day in the Life of an Engineering Firm*, SC24-5352. DEMO1 demonstrates SQL capabilities in the *VM/IS Learning to Use Your System: Working with a Data Base*, SC24-5348.

- DEMO2 (VM/IS-Productivity Facility)

  The Senior Design Engineer uses the DEMO2 userid in the system scenario described in *VM/IS Learning to Use Your System: A Day in the Life of an Engineering Firm*, SC24-5352.

- DEMO3 (VM/IS-Productivity Facility)

  The Junior Design Engineer uses the DEMO3 userid in *VM/IS Learning to Use Your System: A Day in the Life of an Engineering Firm*, SC24-5352.

- DEMO4 (VM/IS-Productivity Facility)

  The Purchasing Officer uses the DEMO4 userid in *VM/IS Learning to Use Your System: A Day in the Life of an Engineering Firm*, SC24-5352. DEMO4 also demonstrates SQL capabilities in *VM/IS Learning to Use Your System: Working with a Data Base*, SC24-5348.

- DIRMAINT (DIRMAINT)

  The DIRMAINT virtual machine manages the CP directory source and associated control files. DIRMAINT makes changes to the CP directory source file so you do not have to manually edit the file. DIRMAINT should be autologged at system IPL time.

- DISKACNT (IPF)

  DISKACNT collects accounting information generated by VM/SP.

- EREP (EREP)

  The IBM Customer Service Representative (CSR) uses EREP while running CPEREP. The privilege class of F lets the CSR clear the error recording area. The maximum virtual size is set at 2 MB so the EREP table can be increased in size if needed.

- FSFCNTRL, FSFTASK1, FSFTASK2, and FSFADMIN (FILE STORAGE)

  These userids correspond to the VM File Storage Facility (VMFSF) that provides disk space and data file management services to CMS users. These services are available to users in a single-host environment and in a multiple-system computing network. Data and disk space can be shared with varying levels of protection selected by individual users. A hierarchical file directory lets users cluster files in a logical manner and increases resolution in the identification of data files.

  FSFTASK1 and FSFTASK2 are "task" virtual machines controlled by FSFCNTRL. You can define additional VMFSF task virtual machines to increase the ability of VMFSF to serve multiple users at the same time.

- GCS

  GCS is a virtual machine supervisor that lets group members share VM/SP systems.

- IIPS (IIPS)

  The IIPS userid controls the instructional system invoked by VM/IPF.

  IIPS minidisks are:

  IIPS 191 used for workspace, registering students, and so on.
  IIPS 193 contains on-line courses.

- IPFSERV (IPF)

  VM/IPF needs the IPFSERV userid to build the CP nucleus. IPFSERV, a disconnected machine, runs Device Support Facilities for IPF DASD management.

- ISPVM (ISPF, ISPF/PDF)

  ISPVM is needed to run ISPF and ISPF/PDF. ISPVM runs in the disconnected state and should be logged on at system IPL time using the AUTOLOG1 virtual machine.

- IVPM1 and IVPM2

  IVPM1 and IVPM2 are directory entries for the IVP virtual machine. They are used to run the initial verification procedure of VM/SP.

- MAINT

  The system programmer traditionally uses the MAINT userid to do system maintenance activities. VM/IPF System Support, RSCS, DIRMAINT, and tailoring dialogs support these functions. If the system programmer does other work on the system, establish another userid for this activity, reserving the MAINT userid for system updates and maintenance only.

  MAINT "owns" all system files. A common product minidisk of the MAINT userid and System Offering is:

  MAINT 19E CMS system extension disk linked by all CMS users and is the location of many optional feature program products

  Refer to Appendix A, "Minidisks Reserved for the MAINT Userid," on page A-1 for descriptions of other MAINT minidisks.

- NETVIEW (NetView)

  NETVIEW is the userid for the NetView product -- an authorized machine for Group Control System. The NETVIEW minidisk is:

  NETVIEW 198 holds VSAM files, must be formatted as a DOS minidisk.

- OLTSEP

  An IBM Customer Service Representative (CSR) uses OLTSEP when running the OnLine Test System Executive Program (OLTSEP). The privilege class of F lets the CSR specify intensive recording mode. Options have been set as needed by OLTSEP.

  The OLTSEP minidisk is:

  OLTSEP 5FF CSR's OLTSEP volume automatically IPLed in the virtual machine.

  Note: This userid is not in the directory for FBA (3370) DASD.

- OPERATNS

  VM/IPF Problem Control Facility (PCF) needs the OPERATNS userid. All IPCS Extended problems are sent to this userid that provides standard IPCS Extended services. All CP system dumps are also sent to OPERATNS, because of the "SYSDUMP=OPERATNS" entry in the DMKSYS ASSEMBLE file.

  OPERATNS minidisks are:

  OPERATNS 191 used as a work disk to process dump files.
  OPERATNS 193 contains one current dump and previous reports generated using the VM/IPF Problem Control Facility dialogs.

  Refer to the *VM/SP Planning Guide and Reference* for more information about the OPERATNS virtual machine.

- OP1 (IPF) and OPERATOR

  The operator uses the OPERATOR and OP1 userids to control the system. If the VM/IPF operation dialogs are run on the OPERATOR userid, system and user messages sent to the operator are displayed only when you press ENTER or otherwise clear the screen. To avoid these interruptions, logon the OP1 userid at a terminal and run the operator dialogs from there. The OPERATOR userid can then be logged onto the console and reserved for system and user messages.

- PDM470 and PDMREM1 (PSF/VM)

  The Printer Driver Machines (PDM), PDM470 and PDREM1, drive the 3800 or 3820 respectively using the data stream and the control files created by the Spool File Conversion Machine (SFCM). The PDM reads the files from the SFCM A-Disk and sends the appropriate commands to the page printer (CCWs for the 3800 or IPDS commands for the 3820). The PDM also processes the fonts needed for the print file. The PDM tracks the document, produces messages, and deletes the spooled print file when the document has printed. The SFCM monitors the PDM

reader queue. When the SFCM detects that the print file was deleted from the PDM, it deletes corresponding files from its A-disk.

- PROCAL (PROFS)

  The calendar data base disconnected virtual machine stores and updates appointment calendars, conference room schedules, and equipment schedules.

  PROCAL minidisks are:

  PROCAL 191 A-disk, contains the PROFILE EXEC, TEXT decks, and control files.
  PROCAL 5FF
  PROCAL 5FE
  PROCAL 5FD
  PROCAL 5FC
  PROCAL 5FB storage disks for calendars and calendar authorization files.

- PRODBM (PROFS)

  The PRODBM disconnected virtual machine provides facilities for storing documents and controlling document security. PRODBM is also used during user enrollment.

  PRODBM minidisks are:

  PRODBM 161 B-disk, contains stored index information for documents on the system.
  PRODBM 191 A-disk, contains TEXT decks and a PROFILE EXEC that initiates the PROFS data base manager and other data base control files.
  PRODBM 5FD third document storage disk.
  PRODBM 5FE second document storage disk.
  PRODBM 5FF first document storage disk.

- PROMAIL (PROFS)

  The PROMAIL disconnected virtual machine provides facilities for delivery and receipt of all documents.

  PROMAIL minidisks are:

  PROMAIL 151 contains information about incoming and outgoing documents.
  PROMAIL 191 A-disk, contains ZIPMAIL module, buck slips, and remote locations files (REMLOCAD FILE and REMLOCLV FILE) and a PROFILE EXEC that initiates the PROFS distribution manager.

- PSFMAINT (PSF/VM)

  The PSFMAINT userid is defined for maintaining PSF. PSFMAINT minidisks are:

  PSFMAINT 191 A-disk, contains PSF programs
  PSFMAINT 193 contains PSF resources
  PSFMAINT 194 contains PSF fonts

  Format all three disks to your installation's standards.

- PVM (PVM)

  VM/Pass-Through Facility runs in the PVM virtual machine. This program lets VM display station users interactively access the local VM system and remote systems. Remote systems can be other VM systems (with or without PVM installed), or they can be non-VM systems.

- RSCS (RSCS version 1)

  RSCS V1 runs in a single virtual machine as a telecommunications subsystem for the VM/SP system. It needs a standard VM/SP virtual machine with telecommunications facilities attached to that virtual machine. Regular operation can be conducted without operator intervention with RSCS running disconnected.

- RSCSV2 (RSCS version 2)

  RSCS V2 runs in a virtual machine and relies on the VM/SP Group Control System (GCS) for supervisor services. The RSCS V2 virtual machine must be defined as part of a GCS virtual machine group. For a Systems Network Architecture (SNA) environment, VTAM must also be defined as part of the same GCS virtual machine group. Regular operation can be conducted without operator intervention with RSCS V2 running disconnected.

- SFCM1 (PSF/VM)

  The Spool File Conversion Machine (SFCM) converts the information from the print file and resource files sent by the PSF command to the appropriate data stream, either channel control words (CCWs) for the 3800 or intelligent-printer data stream (IPDS) commands for the 3820. Line-format print files are converted first to composed-text print files and then into CCWs or IPDS commands. It also builds a set of control files that contain additional information needed by the Printer Driver Machine (PDM) to do the print processing. The converted spool files and control files are stored on the SFCM A-disk for PDM to use.

- SMART (VM/RTM)

  SMART is a userid for VM/RTM, a real-time monitor and diagnostic tool for short term monitoring, analysis, and problem solving. Run SMART as a disconnected virtual machine so many users can share a single copy of the program. SMART should be autologged by AUTOLOG1 when the system is IPLed, and the SMINIT EXEC should be executed as the last procedure in AUTOLOG1.

- SQLDBA (SQL/DS)

  The SQL/DS data base machine is called the SQLDBA machine but can be defined with any valid machine userid. The SQLDBA machine identifies the SQL/DS data base machine that owns the SQL/DS service and production minidisks and is used during SQL/DS installation processing.

  SQLDBA minidisks are:

  SQLDBA 191 A-disk, work disk for the SQL/DS data base machine.
  SQLDBA 193 SQL/DS service minidisk used for SQL/DS data base installation, generation, and maintenance activities.
  SQLDBA 195 SQL/DS production minidisk used for all SQL/DS operations.
  SQLDBA 200 directory disk for the SQL/DS starter data base.
  SQLDBA 201 log disk for the SQL/DS starter data base.
  SQLDBA 202 data disk for the SQL/DS starter data base.

- SQLUSER (SQL/DS)

  SQLUSER is a SQL/DS user machine. You need one SQL/DS user machine to complete SQL/DS installation; however, you can define additional user machines.

- SYSADMIN (PROFS)

  The System Administrator, who is repsonsible for the maintenance and administration of the PROFS system, uses the SYSADMIN userid.

  SYSADMIN minidisks are:

  SYSADMIN 191 A-disk, contains the PROFLOAD EXEC, the Memo to Users, and execs and modules to support the PROFS Administrator function.
  SYSADMIN 298 contains PROFS help screens, messages, and source code.
  SYSADMIN 398 contains PROFS TEXT decks, execs, and files used for the PROFS disconnected virtual machines and for loading PROFS in a DCSS.
  SYSADMIN 399 contains PROFS user routines and supporting files and the remote locations file (REMLOC FILE).

- SYSDUMP1 (IPF)

  The SYSDUMP1 virtual machine handles the DASD volume backups, either on a specified schedule, or as requested by an administrator or the operator. This virtual machine also provides services for restoring a specified user's CMS files from backup tapes.

- TSAFVM

  The communication server for TSAF runs in the TSAFVM virtual machine. It needs a standard VM/SP virtual machine with communication facilities attached or dedicated to that virtual machine. For an abend, TSAF tries to restart itself using the CONCEAL option in the TSAFVM directory entry, so the virtual machine's PROFILE EXEC should contain a RUNTSAF command.

  TSAF can be autologged by AUTOLOG1. Regular operation can be conducted without operator intervention with TSAFVM running disconnected.

  Refer to the *VM/SP Transparent Services Access Facility Reference* for more information about TSAF.

- VMARCH (VMBACKUP-MS)

  The VMARCH service virtual machine runs the VMarchive subsystem of VMBACKUP-MS. VMARCH controls CMS file archival and recall requests generated by users.

  VMARCH minidisks are:

  VMARCH 100 STAGE minidisk used for temporary storage.
  VMARCH 101 STAGE minidisk used for temporary storage.
  VMARCH 191 A-disk, contains database and transaction files.
  VMARCH 193 contains the executable program code.
  VMARCH 200 ONLINE minidisk used for permanent storage.

- VMASMON (AS)

  This userid is the monitor userid that makes sure the number of users that can sign on and gather statistics about AS usage does not exceed the maximum. VMASMON runs as a disconnected virtual machine and should be autologged by the AUTOLOG1 virtual machine when the system is IPLed.

  VMASMON 191 Used as a work disk and contains the statistics gathered.

- VMASSYS (AS)

  AS (VM) needs this userid for the installation of AS. It needs Class E authority to allow saving of the shared segment.

  VMASSYS 191 Installation disk that contains files needed to build and save the shared segment for AS, the monitor program, and the national language support modules. Also used as a work disk and to save shared segment maps.
  VMASSYS 391 System disk that contains files needed for AS execution. All AS users must link to it.
  VMASSYS 392 Test disk that contains files needed to run the installation verification test and must be linked by the test userid.
  VMASSYS 393 Holds IPCS dump during installation and maintenance.

- VMASTEST (AS)

  This userid runs the AS installation verification test.

  VMASTEST 191 Work disk.

- VMBACKUP (VMBACKUP-MS)

  The VMBACKUP service virtual machine handles the minidisk dump and restore function for VMBACKUP-MS. VMBACKUP maintains an on-line catalog that monitors what data is dumped, when it is dumped, and where it is stored.

  VMBACKUP minidisks are:

  VMBACKUP 191 A-disk, contains the executable program code.
  VMBACKUP 192 "work" disk used during restore operations.
  VMBACKUP 193 contains the tape spool files.
  VMBACKUP 194 contains the on-line catalog.

- VMBATCH, VMBATCH1, and VMBATCH2 (VMBATCH)

  The program product VM Batch Subsystem uses these userids to provide a facility for scheduling, initiating, and monitoring batch jobs in a VM/CMS environment. A supervisory virtual machine that dispatches and monitors other virtual machines in which the batch jobs are processed controls the subsystem.

  The VM batch subsystem monitor receives and responds to jobs and commands from users. When a task machine is available to process a job, the monitor selects an appropriate batch job and starts the job in the machine.

  VMBATCH1 and VMBATCH2 are virtual machines that run user jobs. They are controlled by VMBATCH, which determines the sequence of

jobs started, monitors job activity, and processes commands from users. Defining additional batch job-processing virtual machines will make it possible to run more jobs at the same time. An AUTOLOG command from VMBATCH starts the job-processing virtual machines when they are to process jobs. The task machines are not logged on when they are not processing jobs. VMBATCH always runs disconnected.

- VMBSYSAD (VMBACKUP-MS)

  The VMBSYSAD userid provides administrative functions for VMBackup. VMBSYSAD usually generates and submits VMBackup dump jobs. In addition, userid VMBSYSAD is authorized to issue privileged subcommands to the VMArchive subsystem.

  VMBSYSAD minidisks are:

  VMBSYSAD 191 A-disk, contains the job submission exec.
  VMBSYSAD 192 contains files for public use.

- VMMAP (VMMAP)

  Installations that want to take advantage of the VM/SP performance monitoring capabilities for long-term capacity planning and system tuning should install the VM Monitor Analysis Program on the VMMAP virtual machine to do the data reduction and report processing. VMMAP can run as a disconnected virtual machine if you want to establish automatic data collection.

- VMTAPE (VMTAPE-MS)

  The VMTAPE virtual machine manages tape volumes and tape drives and services tape mount requests. It usually runs disconnected.

  VMTAPE minidisks are:

  VMTAPE 191 A-disk, contains the executable program code.
  VMTAPE 200 contains the Tape Management Catalog (TMC).
  VMTAPE 300 contains the Audit File.

- VMTLIBR (VMTAPE-MS)

  The VMTLIBR userid does administrative and control functions for VMTape. Facilities are available for making backups of the TMC and for generating reports. The VMTLIBR userid is also authorized to do privileged operations for VMTape.

  VMTLIBR minidisks are:

  VMTLIBR 191 A-disk, contains the executable program code.
  VMTLIBR 192 used for storing backups of the TMC.

- VMUTIL (IPF)

  The VMUTIL virtual machine can handle tasks that occur repeatedly on a schedule you select. Such tasks are scheduling periodic DASD volume backups and repeating a tape mount request periodically.

- VM3812 (VM3812)

  The VM3812 userid should be autologged. VM3812 minidisks are:

  VM3812 191 PROFILE and temporary files.
  VM3812 192 Executable code and files users cannot tailor.
  VM3812 193 Fonts and files users can tailor.

- VTAM (VTAM)

  The VTAM userid is a Systems Network Architecture (SNA) access method supported in the native VM/SP environment by the Group Control System (GCS). VTAM provides a proven set of facilities and support for using certain terminal devices as virtual machine operator consoles. VM SNA Console Support (VSCS), an integrated component of VTAM, provides this support.

Shared Segment Information

The following table describes the sample DMKSNT layout for optional feature program products. It lists the name of the saved system or Discontiguous Saved Segment (DCSS), the numbers of the pages to be saved for this system or segment (SYSPGNM), the segment numbers to be shared (SYSHRSG), and the overlaps between the saved systems and Discontiguous Saved Segments of different optional feature program products. You cannot run overlapping saved systems and Discontiguous Saved Segments at the same time in the same virtual machine.

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	Overlaps DCSS
APL2	AP2R20S1	1536-1791	96-111	CFSearch/370	DUASEG
				DisplayWrite/370	DW370
				IBM BASIC	BASSEG
					BLISEG
				QMF	QMF210E
	AP2SM2	1792-1919	112-119	CSP/AD	DCBPMS00
					DCALIS00
				CSP/AE	DCBDZMOD
				CSP/Q	DQNINIT
				IIPS	IISDCSS
				PSAF	PSAFDCSS
				QMF	QMF210E
AS	DASSSV40	1184-1535	74-95	CSP/AE	DCFMODS
				GCS	GCS
				ISPF/PDF	ISRDCSS
CFSearch/370	DUASEG	1536-1727	96-107	APL2	AP2R20S1
				DisplayWrite/370	DW370
				IBM BASIC	BASSEG
					BLISEG
				QMF	QMF210E
CSP/AD	DCBPMS00	1872-1903	117,118	APL2	AP2SM2
				CSP/AE	DCBDZMOD
	DCALIS00	1904-1935	119,120	APL2	AP2SM2
				CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAAPP02	1936-1951	121	CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAAPP05	1952-1967	122	CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAAPP06	1968-1983	123	CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG

Figure 9-2 (Part 1 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
	DCAAPP07	1984-1999	124	CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAAPP09	2000-2015	125	CSP/Q	DQNINIT
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAITF01	2032-2063	127,128	CSP/AD	DCAPPR31
					DCA[illegible]33
					DCB 4
					DCAGEN00
					DCAMAP00
					DCAMAP03
					DCADAT00
					DCADAT10
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAITF02	2064-2111	129-131	CSP/AD	DCAPPR35
					DCAGEN32
					DCAMAP04
					DCAMAP10
					DCADAT20
					DCADAT30
					DCADAT40
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAITF05	2112-2127	132	CSP/AD	DCADAT50
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAPPR31	2032-2047	127	CSP/AD	DCAITF01
					DCBPSG04
					DCAMAP00
					DCADAT00
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG

Figure 9-2 (Part 2 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
	DCAPPR33	2048-2063	128	CSP/AD	DCAITF01
					DCAGEN00
					DCAMAP03
					DCADAT10
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAPPR35	2064-2079	129	CSP/AD	DCAITF02
					DCAGEN32
					DCAMAP04
					DCADAT20
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCBPSG04	2032-2047	127	CSP/AD	DCAITF01
					DCAPPR31
					DCAMAP00
					DCADAT00
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAGEN00	2048-2063	128	CSP/AD	DCAITF01
					DCAPPR33
					DCAMAP03
					DCADAT10
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAGEN32	2064-2095	129,130	CSP/AD	DCAITF02
					DCAPPR35
					DCAMAP04
					DCAMAP10
					DCADAT20
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAGEN62	2128-2143	133	GDQF	EMGDQ40
				PROFS	OFSSEG

Figure 9-2 (Part 3 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
	DCAMAP00	2032-2047	127	CSP/AD	DCAITF01
					DCAPPR31
					DCBPSG04
					DCADAT00
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAMAP03	2048-2063	128	CSP/AD	DCAITF01
					DCAPPR33
					DCAGEN00
					DCADAT10
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCAMAP04	2064-2079	129	CSP/AD	DCAITF02
					DCAPPR35
					DCAGEN32
					DCADAT20
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAMAP10	2080-2095	130	CSP/AD	DCAITF02
					DCAGEN32
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAMPP06	2144-2159	134	GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAMPP09	2160-2175	135	GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAMPP11	2176-2191	136	GDQF	EMGDQ40
				PROFS	OFSSEG
	DCADAT00	2032-2047	127	CSP/AD	DCAITF01
					DCAPPR31
					DCBPSG04
					DCAMAP00
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG

Figure 9-2 (Part 4 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
	DCADAT10	2048-2063	128	CSP/AD	DCAITF01
					DCAPPR33
					DCAGEN00
					DCAMAP03
					DCAUTY01
				GDQF	EMGSU40
				PROFS	OFSSEG
	DCADAT20	2064-2079	129	CSP/AD	DCAITF02
					DCAPPR35
					DCAGEN32
					DCAMAP04
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCADAT30	2096-2111	131	CSP/AD	DCAITF02
					DCADAT40
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCADAT40	2096-2111	131	CSP/AD	DCAITF02
					DCADAT30
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCADAT50	2112-2127	132	CSP/AD	DCAITF05
				GDQF	EMGDQ40
				PROFS	OFSSEG
	DCAUTY01	2032-2063	127,128	CSP/AD	DCAITF01
					DCAPPR31
					DCAPPR33
					DCBPSG04
					DCAGEN00
					DCAMAP00
					DCAMAP03
					DCADAT00
					DCADAT10
				GDQF	EMGSU40
				PROFS	OFSSEG
CSP/AE	DCFMODS	1216-1263	76-78	AS	DASSSV40
				GCS	GCS
	DCBDZMOD	1872-1903	117,118	APL2	AP2SM2
				CSP/AD	DCBPMS00

Figure 9-2 (Part 5 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
CSP/Q	DQNINIT	1904-2031	119-126	APL2	AP2SM2
				CSP/AD	DCALIS00
					DCAAPP02
					DCAAPP05
					DCAAPP06
					DCAAPP07
					DCAAPP09
				GDQF	EMGSU40
				PROFS	OFSSEG
DCF	DSMSEG3	2208-2303	138-143	SQL/DS	SQLISQL
				VTAM	VTAM
DisplayWrite/370	DW370	1536-1791	96-111	APL2	AP2R20S1
				CFSearch/370	DUASEG
				IBM BASIC	BASSEG
					BLISEG
				QMF	QMF210E
GASP	GAASEG	2720-2767	170-172	GDDM/VM	ADMASS00
GDDM-IMD	ADMIM000	3488-3599	218-224	VSE/VSAM	CMSDOS
GDDM-PGF	ADMPG000	3040-3487	190-217	VSE/VSAM	CMSAMS
					CMSVSAM
					CMSBAM
GDDM/VM	ADMASS00	2304-2847	144-177	GASP	GAASEG
				PSF	APRPSFCC
					APRSFCMC
					DCKVTBL
					APRCALLV
				SQL/DS	SQLSQLDS
					SQLXRDS
				VS/FORTRAN	DSSVFORT

Figure 9-2 (Part 6 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
GDQF	EMGSU40	1920-2063	120-128	CSP/AD	DCALIS00
					DCAAPP02
					DCAAPP05
					DCAAPP06
					DCAAPP07
					DCAAPP09
					DCAITF01
					DCAPPR31
					DCAPPR33
					DCBPSG04
					DCAGEN00
					DCAMAP00
					DCAMAP03
					DCADAT00
					DCADAT10
					DCAUTY01
				CSP/Q	DQNINIT
				PROFS	OFSSEG
	EMGDQ40	2064-2191	129-136	CSP/AD	DCAITF02
					DCAITF05
					DCAPPR35
					DCAGEN32
					DCAGEN62
					DCAMAP04
					DCAMAP10
					DCAMPP06
					DCAMPP09
					DCAMPP11
					DCADAT20
					DCADAT30
					DCADAT40
					DCADAT50
				PROFS	OFSSEG
IBM BASIC	BASSEG	1536-1663	96-103	APL2	AP2R20S1
				CFSearch/370	DUASEG
				DisplayWrite/370	DW370
				QMF	QMF210E
	BLISEG	1664-1759	104-109	APL2	AP2R20S1
				CFSearch/370	DUASEG
				DisplayWrite/370	DW370
				QMF	QMF210E

Figure 9-2 (Part 7 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	DCSS
IIPS	IISDCSS	1792-1855	112-115	APL2	AP2SM2
				PSAF	PSAFDCSS
				QMF	QMF210E
ISPF	ISPDCSS	2848-3007	178-187		
ISPF/PDF	ISRDCSS	1280-1535	80-95	AS	DASSSV40
PROFS	OFSSEG	1920-2191	120-134	CSP/AD	DCALIS00
					DCAAPP02
					DCAAPP05
					DCAAPP06
					DCAAPP07
					DCAAPP09
					DCAITF01
					DCAITF02
					DCAITF05
					DCAPPR31
					DCAPPR33
					DCAPPR35
					DCBPSG04
					DCAGEN00
					DCAGEN32
					DCAGEN62
					DCAMAP00
					DCAMAP03
					DCAMAP04
					DCAMAP10
					DCAMPP06
					DCAMPP09
					DCAMPP11
					DCADAT00
					DCADAT10
					DCADAT20
					DCADAT30
					DCADAT40
					DCADAT50
					DCAUTY01
				CSP/Q	DQNINIT
				GDQF	EMGSU40
					EMGDQ40
PSAF	PSAFDCSS	1792-1871	112-116	APL2	AP2SM2
				IIPS	IISDCSS
				QMF	QMF210E

Figure 9-2 (Part 8 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	Overlaps DCSS
PSF/VM	APRPSFCC	2304-2319	144	GDDM/VM SQL/DS VS/FORTRAN	ADMASS00 SQLSQLDS DSSVFORT
	APRSFCMC	2320-2383	145-148	GDDM/VM SQL/DS VS/FORTRAN	ADMASS00 SQLSQLDS DSSVFORT
	DCKVTBL	2384-2415	149,150	GDDM/VM SQL/DS VS/FORTRAN	ADMASS00 SQLSQLDS DSSVFORT
	APRCALLV	2416-2671	151-166	GDDM/VM SQL/DS VS/FORTRAN	ADMASS00 SQLSQLDS SQLXRDS DSSVFORT
QMF	QMF210E	1536-1871	96-116	APL2 CFSearch/370 DisplayWrite/370 IBM BASIC IIPS PSAF	AP2R20S1 AP2SM2 DUASEG DW370 BASSEG BLISEG IISDCSS PSAFDCSS
SQL/DS	SQLRMGR	2192-2207	137		
	SQLISQL	2208-2303	138-143	DCF VTAM	DSMSEG3 VTAM
	SQLSQLDS	2304-2511	144-156	GDDM/VM PSF VS/FORTRAN	ADMASS00 APRPSFCC APRSFCMC DCKVTBL APRCALLV DSSVFORT
	SQLXRDS	2512-2719	157-169	GDDM/VM PSF VS/FORTRAN	ADMASS00 APRCALLV DSSVFORT
VM/IS-Productivity Facility	ESCMDCSS	1168-1183	73	GCS	GCS

Figure 9-2 (Part 9 of 10). SNT Planning Input

Product	Saved System or DCSS	Page Number	Saved Segment Number	Overlaps Product	Overlaps DCSS
VSE/VSAM	CMSAMS	3072-3215	192-200	GDDM-PGF	ADMPG000
	CMSVSAM	3216-3327	201-207	GDDM-PGF	ADMPG000
	CMSBAM	3328-3375	208-210	GDDM-PGF	ADMPG000
	CMSDOS	3584-3599	224	GDDM-IMD	ADMIM000
VS/FORTRAN	DSSVFORT	2304-2639	144-164	GDDM/VM PSF	ADMASS00 APRPSFCC APRSFCMC DCKVTBL APRCALLV
				SQL/DS	SQLSQLDS SQLXRDS
	FTNLIB10	3008-3039	188,189		
VTAM	VTAM	2208-2303	138-143	DCF SQL/DS	DSMSEG3 SQLISQL

Figure 9-2 (Part 10 of 10). SNT Planning Input

Using the SNTINFO EXEC to Get DCSS Information

Use the SNTINFO command to get DCSS information directly from CP.

The format of the SNTINFO command is:

SNTINFO	*dcssname* [**(STACK[)]**]

dcssname
is the Discontiguous Saved Segment name that you want information about.

STACK
specifies that the DCSS information is to be put on the stack (LIFO) instead of being displayed on the terminal. No error messages are issued when the STACK option is specified.

Notes:

1. *You can invoke SNTINFO from the CMS command line, or it can be called from a REXX or EXEC2 exec.*

2. *If you do not specify the STACK option, the following line (8 tokens) is displayed on the terminal:*

```
START(HEX): <start>  END(HEX): <end>  SIZE(HEX): <size>  CC: <cc>
```

If you specify the STACK option, the following line (4 tokens) is pushed LIFO onto the stack:

```
<start> <end> <size> <cc>
```

where:

start *is the DCSS start load address in hexadecimal*
end *is the DCSS end load address in hexadecimal*
size *is the DCSS size in hexadecimal*
cc *is the DCSS condition code from the CP DIAGNOSE command.*

Chapter 10. Messages and Panels

Messages

002E **File** *fn ft fm* **not found**

Refer to *VM/SP System Messages and Codes*.

003E **Invalid option:** *option*

Refer to *VM/SP System Messages and Codes*.

005E **No** *option* **specified**

Refer to *VM/SP System Messages and Codes*.

007E **File** *fn ft fm* **is not fixed, 80-character records**

Refer to *VM/SP System Messages and Codes*.

037E **Disk** *mode* **is accessed as read/only**

Refer to *VM/SP System Messages and Codes*.

054E **Incomplete fileid specified**

Refer to *VM/SP System Messages and Codes*.

066E *option1* **and** *option2* **are conflicting options**

Refer to *VM/SP System Messages and Codes*.

069E **Disk** *mode* **not accessed**

Refer to *VM/SP System Messages and Codes*.

283E **The *name* DCSS could not be found; condition code *cc*, return code *rc* from FINDSYS**

Refer to *VM/SP System Messages and Codes*.

336E **No DCSS name specified**

Explanation: The SNTINFO command was invoked with no parameters, but the name of a Discontiguous Saved Segment (DCSS) is needed.

System Action: RC=24
Command execution stops.

User Response: Invoke the command with a valid DCSS name.

337E **DCSS name *name* is longer than eight characters**

Explanation: The name of a Discontiguous Saved Segment (DCSS) cannot be more than eight characters long.

System Action: RC=24
Command execution stops.

User Response: Invoke the command with a valid DCSS name.

648E **Userid *userid* was not found**

Refer to *VM/SP System Messages and Codes*.

657E **Undefined PFkey/PAkey**

Refer to *VM/SP System Messages and Codes*.

833E **Unable to process *fn ft fm***

Explanation: A REXX coding error was detected while interpreting the file.

System Action: RC=64
Command execution stops.

User Response: Make sure the file has not been modified, and report the problem to the IBM Support Center.

834E **Product number *prodnum* requires a feature identification code; correct your input and run DIRECGEN again for all selected products**

Explanation: *prodnum* needs a feature identification code for DIRECGEN to distinguish between components of the product that you can install separately.

System Action: RC=8
Command execution stops.

User Response: Refer to the FEATURE$ PRODUCTS file for a list of valid product numbers and feature identification codes. Invoke the command with the correct input.

835W **Additional manual updates must be made to *fn ft fm*; refer to DIRECGEN HISTORY file for information**

Explanation: Manual minidisk expansion is needed, or additional control statements must be added to a userid. DIRECGEN does not expand the size of existing minidisks or add control statements that might result in duplication. Manual updates might be needed.

System Action: RC=4
Command execution continues.

User Response: Refer to the DIRECGEN HISTORY file for more information about the updates.

836E ***prodnum* is not a System Offering product number; correct your input and run DIRECGEN again for all selected products**

Explanation: DIRECGEN works only for System Offering products listed in the FEATURE$ PRODUCTS file. *prodnum* is an invalid product or is not a System Offering product.

System Action: RC=8
Command execution stops.

User Response: Refer to the FEATURE$ PRODUCTS file for a list of valid System Offering product numbers. Invoke the command with the correct input.

837E **Incorrect feature identification code *idcode* specified for product number *prodnum*; correct your input and run DIRECGEN again for all selected products**

Explanation: You entered a feature identification code that is either invalid or not needed for this product number.

System Action: RC=8
Command execution stops.

User Response: Refer to the FEATURE$ PRODUCTS file for a list of valid product numbers and feature identification codes. Invoke the command with the correct input.

838E **Insufficient DASD space for required minidisks; refer to DIRECGEN HISTORY file for information**

Explanation: Not enough space for the minidisks of selected products is available on the DASD specified in the DIRECGEN EXTENTS file.

System Action: RC=16
Command execution stops.

User Response: Refer to the DIRECGEN HISTORY file for an estimate of additional DASD needed. Then, specify more DASD space in the DIRECGEN EXTENTS file, or invoke the command with a smaller product set.

839W **Directory resources have already been added for *prodnum idcode*; product will be bypassed and processing will continue**

Explanation: *prodnum idcode* has been used in a previous successful execution of DIRECGEN. Therefore, DIRECGEN will not try to add the directory requirements for this product.

System Action: Command execution continues.

User Response: Refer to the DIRECGEN HISTORY file for a list of products that have directory resources allocated.

840I **New directory successfully created:** *fn ft fm*

Explanation: A new CP directory has been created containing the directory requirements for the selected products. If you also received the message 835W, you might have to make additional manual updates as recorded in the DIRECGEN HISTORY file.

System Action: Command execution continues.

User Response: No user action is required.

841I **Old** *fn ft fm* **file renamed VMOLD DIRECT** *fm*

Explanation: The input CP directory has been saved as a backup in the file VMOLD DIRECT.

System Action: Command execution continues.

User Response: No user action is required.

889E **Product number** *prodnum* **was not found in PROGPROD PARMLIST file**

Explanation: *prodnum* was not found in the PROGPROD PARMLIST file.

System Action: RC=64
Command execution stops.

User Response: Make sure the PROGPROD PARMLIST and the FEATURE$ PRODUCTS files have not been modified, and report the problem to the IBM Support Center.

890R **Enter the name of the file containing the list of Program Products to be installed:**

Explanation: The product selection panel cannot be displayed because the terminal is in line mode. You must create a file containing the products that need directory resources. This file must be in the same format as the FEATURE$ PRODUCTS file.

System Action: The system waits for a response.

User Response: If a product list file has been created, enter the filename and filetype. If you omit filemode, the default is '*'. If a product list file has not been created, enter a null line to exit and enter HELP DIRECGEN to get more information about running DIRECGEN in line mode.

891E **Userid *userid* is required before adding the directory requirements for product number *prodnum idcode***

Explanation: The CP directory file does not contain a userid that is a prerequisite for adding the directory requirements of this product.

System Action: RC=64
Command execution stops.

User Response: Make sure you select all prerequisite products, then invoke the command. If the userid reported was in the IBM supplied sample CP directory file, add it to the input CP directory and invoke the command again.

1001E **Disk *mode(vdev)* containing *fn ft* file is read/only**

Explanation: The file *fn ft* must be updated during command execution. It cannot be updated because the minidisk containing the file is accessed read/only.

System Action: RC=36
Command execution stops.

User Response: Make sure the minidisk is accessed R/W, and invoke the command again.

1002E **{Existing|Common} disk *userid vdev* can not have DASD specific sizing**

Explanation:

(For an existing disk) *userid vdev* is already in the CP directory. Because of the specific requirements of the disk DIRECGEN has to create, this disk cannot be in the CP directory before running DIRECGEN with the same product selection.

(For a common disk) There is an error in the PROGPROD PARMLIST file. Conflicting requirements have been specified for the disk identified.

System Action: RC=64
Command execution stops.

User Response:

(For an existing disk) Change the address of the existing minidisk and invoke the command again.

(For a common disk) Make sure the PROGPROD PARMLIST file has not been modified, and report the problem to the IBM Support Center.

1003E Error found in line *nn* of *fn ft* file

Explanation: There is an error in a required DIRECGEN input file.

System Action: RC=64
Command execution stops.

User Response: Make sure the file has not been modified and report the problem to the IBM Support Center.

1004E Insufficient DASD space for product number *prodnum*; refer to DIRECGEN HISTORY file for information

Explanation: There is not enough DASD space available on the DASD extents specified in the DIRECGEN EXTENTS file for *prodnum*.

System Action: RC=16
Command execution stops.

User Response: Look in the DIRECGEN HISTORY file for an approximation of the additional DASD space needed. Specify more DASD space for this product in the DIRECGEN EXTENTS file, or invoke the command without selecting this product.

1005E Sufficient space is required on at least two packs for product number *prodnum*

Explanation: Available DASD space on at least two packs must be listed in the DIRECGEN EXTENTS file for *prodnum*. Two minidisks for this product cannot be on the same DASD volume.

System Action: RC=16
Command execution stops.

User Response: Add an extent for an additional pack in the DIRECGEN EXTENTS file, or invoke the command without selecting this product.

1006E **Invalid entry *entry* on line *nn* of DIRECGEN EXTENTS file**

Explanation: There is an error in the DIRECGEN EXTENTS file. There might be an unsupported device or an invalid START or SIZE entry. Supported devices are listed in the comments in the DIRECGEN EXTENTS file. The START entry must be a positive integer, and the SIZE entry must be an asterisk or a positive integer.

System Action: RC=64
Command execution stops.

User Response: Correct the DIRECGEN EXTENTS file and invoke the command again.

1007I **Directory generation in progress...**

Explanation: All input parameters and options have been verified, and the directory generation process is underway.

System Action: Command execution continues.

User Response: No user action is required.

1008I **The DIRECGEN HISTORY file is now being updated...**

Explanation: Directory processing has ended, and the DIRECGEN HISTORY file is being updated with status information.

System Action: Command execution continues.

User Response: No user action is required.

1009I **A console file is being created containing information which was not recorded in the DIRECGEN HISTORY file**

Explanation: There was an error while writing to the DIRECGEN HISTORY file. Information you might have lost is sent to the virtual reader in a console file.

System Action: Command execution continues.

User Response: For a record of the DIRECGEN execution history, refer to the DIRECGEN HISTORY file and the newly created console file. After correcting the write error to the disk containing the DIRECGEN HISTORY file, attach the console file to the history file to maintain a complete history.

1010W **Userid *userid* for product number *prodnum* already exists; manual updates may be required for this userid; refer to DIRECGEN HISTORY file for information**

Explanation: A userid to be created for a selected product is already in the input CP directory. DIRECGEN did not update the control statements for this userid.

System Action: RC=4
Command execution continues.

User Response: The required control statements for the product have been recorded in the DIRECGEN HISTORY file. Examine these entries, and update the existing userid control statements in the CP directory as needed.

1011E **Error occurred creating directory *fn ft fm*; return code *rc* from the CMS COPYFILE command**

Explanation: There might not be sufficient free space on the disk containing the CP directory.

System Action: RC=100
Command execution stops.

User Response: Clear some disk space if there was not enough. Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the COPYFILE command.

1012E **Error occurred renaming *fn ft fm* to VMOLD DIRECT *fm*; return code *rc* from the CMS RENAME command**

Explanation: The RENAME command failed. The name of the input CP directory cannot be VMOLD DIRECT. A backup of the input directory is stored as VMOLD DIRECT during command execution.

System Action: RC=100
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the RENAME command.

1013E **Pack *label* must be specified in the DIRECGEN EXTENTS file for product number *prodnum***

Explanation: An entry for the reported pack must be in the DIRECGEN EXTENTS file for this product.

System Action: RC=64
Command execution stops.

User Response: Add an entry for this pack in the DIRECGEN EXTENTS file and code a 0 in the size field if no minidisks should reside on this pack.

1014E **No new product selections were made**

Explanation: You did not make any new product selections, or all the products you selected have been processed in a previous execution of DIRECGEN as indicated by the message 839W.

System Action: RC=8
Command execution stops.

User Response: Refer to the DIRECGEN HISTORY file for a list of products that have directory resources allocated. Invoke the command with the correct input.

1015E **EXECIO error *rc* reading file *fn ft fm* from disk**

Explanation: There was an I/O error reading *fn ft fm*.

System Action: RC=100
Command execution stops.

User Response: If the disk is read/only, make sure it has not been modified since the last access, and try accessing the disk again. Refer to the *VM/SP CMS Command Reference* for more information about the EXECIO command.

1016E **EXECIO error *rc* writing file *fn ft fm* on disk**

Explanation: There was an I/O error writing *fn ft fm*.

System Action: RC=100
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the EXECIO command.

1017W

```
*******************************************
* Logon passwords have been printed; please *
* secure listing to avoid a security exposure *
*******************************************
```

Explanation: A list of the logon passwords has been sent to the printer. If the printer is not in a secure area, the list of passwords is a security hazard.

System Action: Command execution continues.

User Response: Make sure the listing is secure.

1018I *nn* **userids still have the logon password NOLOG**

Explanation: A userid with the logon password NOLOG cannot be used until you change the password.

System Action: Command execution continues.

User Response: Invoke the command to change any remaining NOLOG userids.

1019I **The directory source file has been updated with new logon passwords; use the CMS DIRECT command to place these changes online**

Explanation: A new CP directory file has been created. This directory takes effect when you invoke the CMS DIRECT command.

System Action: Command execution continues.

User Response: Invoke the DIRECT command to activate the new directory.

1020R **The last logon password in the list has been updated; please enter EXECUTE to place the passwords in the CP directory file or enter QUIT to exit without updating the directory**

Explanation: You are using PASSMOD on a line-mode terminal. You have had a chance to update the logon password for each userid in the CP directory file. The new passwords you entered have not been written to the CP directory. They will be written if you enter EXECUTE on the command line.

System Action: The system waits for a response.

User Response: Enter either EXECUTE or QUIT on the command line, but do not abbreviate them.

1021W **Logon passwords will not be saved; press PF3 to quit anyway**

Explanation: A request has been made to cancel all changes entered.

System Action: The system waits for user input from the data entry panel.

User Response: Press PF3 (QUIT) again to exit without updating the CP directory. Any other action lets logon password modification continue.

1022W **Logon passwords will not be saved; enter QUIT to quit anyway or press <ENTER> to continue**

Explanation: You are using PASSMOD on a line-mode terminal. You entered QUIT on the command line. You have entered new logon passwords, but the CP directory has not been updated with the changes.

System Action: The system waits for a response.

User Response: Enter QUIT again to exit without updating the CP directory or press ENTER to continue processing logon passwords.

1023R **If you would like a list of userids and new logon passwords to be printed, enter 'YES'; otherwise, press <ENTER> to continue execution**

Explanation: The list of userids and logon passwords is sent to the virtual printer if you enter YES. Where and how this file is printed depends on the printer spool options in effect before you invoke PASSMOD.

System Action: The system waits for a response.

User Response: Respond YES if you want the list printed. Remember the printed list represents a security exposure if the printer area is not secure.

1024W **PASSMOD must not be used if DIRMAINT is operational**

Explanation: You cannot use PASSMOD if DIRMAINT is used to maintain your CP directory.

System Action: Command execution continues.

User Response: If DIRMAINT is used, use DIRMAINT commands to change the logon passwords.

1025E **No USER control statements exist in *fn ft* file**

Explanation: PASSMOD expects the input file *fn ft* to be in the format of a CP directory. All CP directories must contain USER statements for each userid.

System Action: RC = 24
Command execution stops.

User Response: Invoke the command specifying a valid CP directory file.

1026W **Logon passwords will not be printed; please ensure that they have been recorded**

Explanation: A hard copy list of the logon passwords is not available. Without a record of passwords to be changed, you can have difficulty accessing the system the next time you try to log on.

System Action: Command execution continues.

User Response: Make sure the logon passwords have been recorded if the CP directory is placed on-line.

1027E **Only one option may be specified**

Explanation: You specified two or more options on the command line. You can only specify one.

System Action: RC=24
Command execution stops.

User Response: Invoke the command with a valid option.

1028E **Invalid logon password; please re-enter with no imbedded blanks**

Explanation: You entered the logon password indicated by the cursor on the PASSMOD data entry panel with imbedded blanks. A valid VM/SP logon password cannot contain imbedded blanks.

System Action: The system waits for user input from the data entry panel. The cursor moves to the field needing correction.

User Response: Correct the logon password on the panel.

1029E **Invalid logon password; must be at most 8 characters with no imbedded blanks; please re-enter**

Explanation: A valid VM/SP logon password cannot contain imbedded blanks and can be at most eight characters in length.

System Action: The system waits for a response.

User Response: Enter a valid logon password.

1030R **Userid** *userid* **[for product** *prodnum***] has current logon password** *password***; enter a new password, or press <ENTER> to retain the current password**

Explanation: The new logon password you enter replaces the current logon password in the input CP directory file.

System Action: The system waits for a response.

User Response: Enter a new logon password or press ENTER to avoid changing the current logon password.

1031I **Enter EXECUTE to process or QUIT to exit**

Explanation: You are using PASSMOD on a line-mode terminal. To update the CP directory file with the logon passwords you entered on the command line, you must enter EXECUTE. If QUIT is entered, the CP directory remains unchanged.

System Action: You are prompted to enter a logon password. Message 1030R or 1033R is displayed.

User Response: Respond to the prompt for a new logon password or enter EXECUTE or QUIT. Do not abbreviate EXECUTE or QUIT.

1032I **The following products were selected:**

Explanation: You pressed the PF5 EXECUTE key on the DIRECGEN panel and the list of selected products is displayed.

System Action: Command execution continues.

User Response: Review the list to make sure you selected the correct products.

1033R **Random logon password** *password* **has been generated for userid** *userid* **[for product** *prodnum***]; enter a new password or press <ENTER> to retain the random password**

Explanation: The previous logon password for the userid *userid* was NOLOG. It has been changed to a randomly selected string. You can keep the new logon password or change it again.

System Action: The system waits for a response.

User Response: Enter a new logon password or press ENTER to retain the random password.

1034E **Invalid product specification code *prodspec*; correct your input and run INSTFPP again for all selected products**

Explanation: You passed an invalid product specification code as a parameter to INSTFPP. A product specification code is seven to nine characters long and is a combination of the product number and the feature identification code. If there is no feature identification code for this product, then the product specification code must be exactly seven characters long. You can leave a blank between the product number and the feature identification code.

System Action: RC = 24
Command execution stops.

User Response: Invoke the command with a valid product specification code. Refer to the FEATURE$ PRODUCTS file, Program Directory, or Memo to Users to obtain the valid product specification code.

1035W ***prodnum* must be entered with a feature identification code; *prodnum idcode* was found on the tape**

Explanation: The product *prodnum* has features. Each feature has its own feature identification code. You must specify both the product number and the idcode together on the command line.

System Action: Command execution continues.

User Response: Invoke the command with valid product specification codes.

1036I **Memo to Users, *fn ft*, has been printed**

Explanation: The Memo to Users for the optional feature program product being processed has been spooled to the virtual printer.

System Action: Command execution continues.

User Response: Retrieve the printed file from the real printer if the appropriate spooling options are in effect.

1037W **The following selected product numbers were not found on the tape:** *prodnums*

Explanation: One or more of the products specified on the command line or selected from the panel is not on the tape at virtual address 181. The wrong tape might be mounted, or you did not correctly enter the product specification codes on the command line.

System Action: RC=85
Command execution continues.

User Response: Make sure the correct tape is mounted, and invoke the command with the correct parameters.

1038E **Error printing** *fn ft fm*; **return code** *rc* **from the CMS PRINT command**

Explanation: The PRINT command with the CC option failed while trying to print the Memo to Users.

System Action: RC=27
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the PRINT command.

1039E **Terminal is in line mode, so the INSTFPP user assistance panel cannot be displayed; INSTFPP arguments must be entered on the command line**

Explanation: You must invoke INSTFPP with parameters and/or options if the terminal is in line mode, even if you want all the default options.

System Action: RC=600
Command execution stops.

User Response: Invoke the command with parameters and/or options.

1040E Virtual machine size must be at least 12M for installation

Explanation: Some installation steps might fail if the virtual storage is less than 12M (12288K bytes). A virtual storage size of 12M or greater is needed to run INSTFPP.

System Action: RC=51
Command execution stops.

User Response: Redefine the virtual storage to 12M or greater using the CP DEFINE STORAGE command and re-ipl CMS before you invoke the command again.

1041E Error {scanning|forwarding} tape *vdev*; return code *rc* from the CMS VMFPLC2 command

Explanation: An error occurred while forward spacing or scanning the tape. There is either a hardware problem or a tape error. The tape position is unknown.

System Action: RC=45
Command execution stops.

User Response: Make sure your hardware is working properly and you have the correct tape mounted. Then, invoke the command again. If the problem persists, contact your IBM Customer Service Representative if you suspect a hardware problem or contact the IBM Support Center if you suspect a problem with the tape itself.

1042E Device at virtual address *vdev* does not exist

Explanation: You invoked INSTFPP without specifying the real address of the stacked program product tape.

System Action: RC=40
Command execution stops.

User Response: Attach the tape unit to MAINT as *vdev* or invoke INSTFPP specifying the real address of the tape unit.

1043E **A Product Identifier file is missing on the tape**

Explanation: INSTFPP did not find a required file on the tape. The tape might have been incorrectly positioned during the installation process.

System Action: RC = 87
Command execution stops.

User Response: Make sure the correct tape is mounted and rewound, and invoke the command again. If the problem persists, contact the IBM Support Center.

1044R **Press <ENTER> to continue or type in any character before pressing <ENTER> to return to the panel:**

Explanation: A list of products selected on the DIRECGEN panels has been displayed after pressing the PF5 EXECUTE key. Execution can continue or the panel can be redisplayed.

System Action: The system waits for a response.

User Response: After reviewing the product list for accuracy, respond to the prompt accordingly.

1045E **Error defining temporary work minidisk** *vdev*; **return code** *rc* **from the CP DEFINE command**

Explanation: The CP DEFINE command failed with return code *rc*. If RC = 91, there might be insufficient temporary disk space available on the system.

System Action: RC = 30
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CP Command Reference* for more information about the DEFINE command.

1046I **Virtual device *vdev1* has been redefined to *vdev2***

Explanation: The virtual device at *vdev1* has been moved to another virtual address, either to avoid detaching the original virtual device or because a particular virtual address is needed for the command to continue.

System Action: The virtual address of the device is redefined. Command execution continues.

User Response: No user action is required.

1047I **Starting to process Program Product *prodnum idcode...***

Explanation: INSTFPP has started to process *prodnum idcode.*

System Action: INSTFPP processes the product according to the options specified.

User Response: No user action is required.

1048E **INSTFPP may only be executed from the MAINT userid**

Explanation: Many installation steps need access to MAINT's minidisks and privilege classes. You cannot run INSTFPP from another userid.

System Action: RC=50
Command execution stops.

User Response: Logon to MAINT userid, and invoke the command again.

1049E **Product numbers may not be specified with the *option* option**

Explanation: The user cannot enter product numbers on the command line with the option *option.*

System Action: RC=24
Command execution stops.

User Response: Invoke the command with correct parameters and options.

1050E There were no Program Products found on the tape

Explanation: The tape is correctly mounted and in VMFPLC2 format, but it is not a stacked program product tape for use with INSTFPP.

System Action: RC = 80
Command execution stops.

User Response: Mount the correct tape and invoke the command again.

1051I Now updating the PROD LEVEL P file...

Explanation: INSTFPP updates the PROD LEVEL file with history information after installing an optional feature program product.

System Action: The PROD LEVEL file on MAINT's 319 minidisk is updated with the version, release, modification, and service level of the installed product. Additional installation requirements are also given. And, the return code is given if a severe error occurred during installation exec processing.

User Response: No user action is required.

1052R Do you wish to install this product now (yes or no)?

Explanation: INSTFPP displays this prompt if the options Install and Prompt are active (specified on the command line, from the user data entry panel, or by default).

System Action: INSTFPP waits for a response.

User Response: If you respond YES, the product installation exec is called to install the product, and the Memo to Users is printed if the Memo option is in effect. If you respond NO, INSTFPP bypasses this product, and processing continues with the next product on the tape.

1053R Do you wish to print the Memo to Users now (yes or no)?

Explanation: INSTFPP displays this prompt if the options NOInstall and Prompt are active (specified on the command line or from the user data entry panel).

System Action: INSTFPP waits for a response.

User Response: If you respond YES, the Memo to Users is printed. Otherwise, processing continues with the next product on the tape. The product is not installed in either case.

1054I Now formatting temporary workdisk *vdev* as *mode*...

Explanation: A temporary minidisk is being prepared for the installation process to use. The formatting process can take some time.

System Action: The minidisk is formatted. Command execution continues.

User Response: No user action is required.

1056I Installation EXEC failed...

Explanation: An installation exec called by INSTFPP exited with a return code other than 0, 777, or 888. Product installation was not completed.

System Action: INSTFPP continues processing with the next product on the tape.

User Response: Once INSTFPP finishes processing, look at the PROD LEVEL file on MAINT's 319 minidisk to determine the return code from the failed installation exec, or look at the console file in the virtual reader. Correct the problem and invoke the command again.

1057I Leaving INSTFPP EXEC...

Explanation: INSTFPP has finished processing.

System Action: INSTFPP restores the environment (accessed disks, spool file characteristics) to its previous state and exits.

User Response: No user action is required.

1058E A valid real tape address must be entered

Explanation: The tape unit address specified on the INSTFPP data entry panel is not a valid device address. A valid device address consists of three or four consecutive hexadecimal digits (0-F).

System Action: The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

User Response: Obtain the real address of the tape unit that has the stacked product tape mounted. The command CP QUERY TAPE can be helpful. Type the real address on the data entry panel and press ENTER.

1059E Reply must be "Y" or "N"

Explanation: Only Y (for YES) and N (for NO) are valid entries in the field.

System Action: The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

User Response: Type either Y or N in the appropriate field and press ENTER.

1060E Option must be 1, 2 or 3

Explanation: Only 1, 2, and 3 are valid processing options.

System Action: The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

User Response: Type a 1, 2, or 3 in the appropriate field and press ENTER.

1061E Device at real address *rdev* does not exist

Explanation: The device address you entered on the INSTFPP data entry panel is not the real address of any device. You might have entered a virtual address instead of a real address.

System Action: The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

User Response: Obtain the real address of the tape unit on which the stacked product tape is mounted. The command CP QUERY TAPE can be helpful. Type the real address on the data entry panel and press ENTER.

1062E Device at real address *rdev* is not a tape unit

Explanation: The device address you entered on the INSTFPP data entry panel is not the real address of a tape unit. You might have entered a virtual address instead of a real address.

System Action: The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

User Response: Obtain the real address of the tape unit that has the stacked product tape mounted. The command CP QUERY TAPE can be helpful. Enter the real address on the data entry panel and press ENTER.

1063E **Error attaching device *rdev* as *vdev***

Explanation: The device at real address *rdev* could not be attached as virtual address *vdev*. Make sure the device is not attached to another user and MAINT has user privilege class A or B.

System Action: RC=40

(In panel mode) The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

(In line mode) Command execution stops.

User Response:

(In panel mode) Try to correct the problem using the panel command line to enter CP/CMS commands. Continue processing the panel or press PF3 to QUIT.

(In line mode) Correct the problem and invoke the command again.

Refer to the *VM/SP Operator's Guide* for more information about the CP ATTACH command.

1064E **Error rewinding tape; return code *rc* from the CP REWIND command**

Explanation: The device at virtual address 181 might not be a tape unit or there might be a hardware tape unit error.

System Action: RC=40

(In panel mode) The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

(In line mode) Command execution stops.

User Response:

(In panel mode) Try to correct the problem using the panel command line to enter CP/CMS commands. Continue processing the panel or press PF3 to QUIT.

(In line mode) Correct the problem and invoke the command again.

Refer to the *VM/SP CP Command Reference* for more information about the CP REWIND command.

1065E Rewind not performed; device *vdev* not ready

Explanation: The tape unit is attached as *vdev*, but the CP REWIND command returned the response REWIND NOT PERFORMED. Possible explanations are: the tape is not mounted properly, the tape is not ready, or the tape is still rewinding from a previous CP REWIND *vdev*.

System Action: RC=40

(In panel mode) The cursor moves to the field needing correction. The system waits for user input from the data entry panel.

(In line mode) Command execution stops.

User Response: Make sure the tape unit is operating correctly and the tape is mounted properly.

(In panel mode) Make sure the tape is ready and press ENTER to reprocess the panel or press PF3 to QUIT.

(In line mode) Make sure the tape is ready and invoke the command again.

1066E Return code *rc* from ACCESS *vdev mode*

Explanation: The access of a minidisk needed for command execution failed.

System Action: RC=36
Command execution stops.

User Response: Make sure all required minidisks are attached, and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the ACCESS command.

1067E Return code *rc* from the CMS XEDIT command

Explanation: The attempt to display the user data entry panel failed.

System Action: RC=32
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the XEDIT command.

1068E **Error copying Memo to Users to MAINT 319 P disk; return code *rc* from the CMS COPYFILE command**

Explanation: The COPYFILE command failed with return code *rc*

System Action: RC = 29
Command execution stops.

User Response: Refer to the *VM/SP CMS Command Reference* for more information about the COPYFILE command. Make sure the MAINT 319 disk is linked in read/write mode and has sufficient DASD space and invoke the command again.

1069E **Error formatting temporary work minidisk *vdev*; return code *rc* from the CMS FORMAT command**

Explanation: The FORMAT command failed with return code *rc*

System Action: RC = 35
Command execution stops.

User Response: Correct the problem and invoke the command again. Refer to the *VM/SP CMS Command Reference* for more information about the FORMAT command.

DIRECGEN Panels

```
SODIR01            DIRECGEN PRODUCT SELECTION PANEL          Line 1 of 68
------------------------------------------------------------------------------
         Type an X next to the product(s) you want represented
         in the CP directory.  When you have finished press the
         PF5 key to begin the generation process.

_  5668854          ACF/Network Control Program
_  5664289          ACF/System Support Program
_  5664280          ACF/Virtual Telecommunications Access Method
_  5668899          APL2
_  5668808          Application Prototype Environment
_  5767032          Application System
_  5798RWL          Composition Utility
_  5664329          Contextual File Search/370 for VM/CMS
_  5664296          Cooperative Viewing Facility Version 2
_  5668813          Cross System Product/Application Development
_  5668814          Cross System Product/Application Execution
_  5668918          Cross System Product/Application Query
------------------------------------------------------------------------------
PF1=Help     2=          3=Quit          4=              5=Execute  6=
PF7=         8=Forward  9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-1. DIRECGEN Panel 1

```
SODIR01            DIRECGEN PRODUCT SELECTION PANEL          Line 13 of 68
------------------------------------------------------------------------------
         Type an X next to the product(s) you want represented
         in the CP directory.  When you have finished press the
         PF5 key to begin the generation process.

_  5748XE4          Directory Maintenance
_  5748XXB          Display Management System for CMS
_  5664370          DisplayWrite/370
_  5748XX9          Document Composition Facility
_  5735XXB          Emulation Program
_  5654260          Environmental Recording Editing and Printing Program
_  5668890          Font Library Service Facility
_  5798DFH          FORTRAN Utilities
_  5668801          Graphical Data Display Manager - IMD
_  5668812          Graphical Data Display Manager - PGF
_  5668812    NL    Graphical Data Display Manager - PGF NL Feature
_  5664200          Graphical Data Display Manager Base
------------------------------------------------------------------------------
PF1=Help     2=          3=Quit          4=              5=Execute  6=
PF7=Backward 8=Forward  9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-2. DIRECGEN Panel 2

```
SODIR01             DIRECGEN PRODUCT SELECTION PANEL          Line 25 of 68

------------------------------------------------------------------------------
          Type an X next to the product(s) you want represented
          in the CP directory.  When you have finished press the
          PF5 key to begin the generation process.

_  5664200    NL    Graphical Data Display Manager Base NL Feature
_  5668905          Graphical Display And Query Facility
_  5799AXX          Graphics Attachment Support Programming
_  5799BKE          Host Displaywriter Document Interchange
_  5668996          IBM BASIC/VM
_  5664185          IBM High-Accuracy Arithmetic Subroutine Library
_  5798DTE          IBM 3812 Pageprinter VM Support
_  5668897          Info Center/1
_  5668012          Interactive Instructional Presentation System
_  5664282          Interactive System Productivity Facility
_  5664285          Interactive System Productivity Facility/PDF
_  5664204    01    NetView Volume 1
------------------------------------------------------------------------------
 PF1=Help     2=         3=Quit          4=              5=Execute  6=
 PF7=Backward 8=Forward  9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-3. DIRECGEN Panel 3

```
SODIR01             DIRECGEN PRODUCT SELECTION PANEL          Line 37 of 68

------------------------------------------------------------------------------
          Type an X next to the product(s) you want represented
          in the CP directory.  When you have finished press the
          PF5 key to begin the generation process.

_  5664204    02    NetView Volume 2
_  5664204    03    NetView Volume 3
_  5734PL3          OS PL/1 Optimizing Compiler and Libraries
_  5734LM5          OS PL/1 Transient Library
_  5664293          Overlay Generation Language
_  5664199          Page Printer Formatting Aid
_  5796PNQ          Pascal/VS
_  5664298          PC Bond
_  5664312          Print Services Access Facility
_  5664198    B     Print Services Facility/VM Base
_  5664198    S     Print Services Facility/VM 3800
_  5664198    V     Print Services Facility/VM 3820
------------------------------------------------------------------------------
 PF1=Help     2=         3=Quit          4=              5=Execute  6=
 PF7=Backward 8=Forward  9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-4. DIRECGEN Panel 4

```
SODIR01            DIRECGEN PRODUCT SELECTION PANEL          Line 49 of 68

------------------------------------------------------------------------------
         Type an X next to the product(s) you want represented
         in the CP directory.  When you have finished press the
         PF5 key to begin the generation process.

_  5664309          Professional Office System
_  5668309    PA    Professional Office System Applications Support Feature
_  5668AAA          Query Management Facility
_  5748XP1          RSCS Networking, Version 1
_  5664188          RSCS Networking, Version 2
_  5748XXJ          Structured Query Language/Data System
_  5664191          Virtual Machine Monitor Analysis Program
_  5664319          Virtual Machine/Personal Computer Host Server
_  5746AM2          Virtual Storage Extended/Virtual Storage Access Method
_  5664318          VM/Interactive Productivity Facility
_  5664364          VM Batch Facility
_  5798DMY          VM File Storage Facility
------------------------------------------------------------------------------
 PF1=Help     2=          3=Quit          4=              5=Execute  6=
 PF7=Backward 8=Forward   9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-5. DIRECGEN Panel 5

```
SODIR01            DIRECGEN PRODUCT SELECTION PANEL          Line 61 of 68

------------------------------------------------------------------------------
         Type an X next to the product(s) you want represented
         in the CP directory.  When you have finished press the
         PF5 key to begin the generation process.

_  5748RC1          VM/Pass-through Facility
_  5796PNA          VM Real Time Monitor
_  5664283          VM/Integrated System-Productivity Facility
_  5664291          VMBACKUP Management System
_  5664292          VMTAPE Management System
_  5668958          VS COBOL II
_  5668806          VS FORTRAN
_  5664281          3270 PC File Transfer




------------------------------------------------------------------------------
 PF1=Help     2=          3=Quit          4=              5=Execute  6=
 PF7=Backward 8=          9=Sort(prodid) 10=Sort(desc)   11=        12=Cursor

====>
```

Figure 10-6. DIRECGEN Panel 6

PASSMOD Panel

```
SOPAS01                PASSWORD MODIFICATION PANEL              Page 1 of 1

--------------------------------------------------------------------------------
          Type the new password for the userid(s) that you wish to
          change in the 'New pw' field.  Press the PF5 key to update
          the passwords in the CP directory file.

 Userid     Current pw  New pw     Prodid    Description

 AUTOLOG1   NOLOG       ........
 CMSBATCH   NOLOG       ........
 CMSUSER    NOLOG       ........
 EREP       NOLOG       ........
 GCS        NOLOG       ........
 IVPM1      NOLOG       ........
 IVPM2      NOLOG       ........
 MAINT      CPCMS       ........
 OLTSEP     NOLOG       ........
 OPERATNS   NOLOG       ........
--------------------------------------------------------------------------------
 PF1=Help     2=Print    3=Quit          4=Random        5=Execute 6=
 PF7=         8=         9=Sort(prodid) 10=Sort(userid) 11=       12=Cursor

 ====>
```

Figure 10-7. PASSMOD Panel

Note: Additional userids may also be displayed depending on the content of the input CP directory.

INSTFPP Panels

```
 SOINS01            INSTFPP INSTALLATION OPTIONS
------------------------------------------------------------------------------

If appropriate, change any defaults and then press the ENTER key.

       Real tape address (will be attached as 181):              ____
       Process all products on the tape (Y/N)?                    Y
       Be prompted before each product is processed (Y/N)?        Y

       Alternatives:
              (1) Install the product(s) and print the memo(s)
              (2) Only print the product memo(s)
              (3) Only install the product(s)
                                          Enter (1, 2, or 3):    1




------------------------------------------------------------------------------
 PF1=Help      2=          3=Quit         4=           5=          6=
 PF7=          8=          9=             10=          11=         12=Cursor

====>
```

Figure 10-8. INSTFPP Panel 1

```
 SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 1 of 68
------------------------------------------------------------------------------
          Type an X next to the product(s) you want to install
          When you have  finished,  press the PF5 key to begin
          the installation process.

_  5668854          ACF/Network Control Program
_  5664289          ACF/System Support Program
_  5664280          ACF/Virtual Telecommunications Access Method
_  5668899          APL2
_  5668808          Application Prototype Environment
_  5767032          Application System
_  5798RWL          Composition Utility
_  5664329          Contextual File Search/370 for VM/CMS
_  5664296          Cooperative Viewing Facility Version 2
_  5668813          Cross System Product/Application Development
_  5668814          Cross System Product/Application Execution
_  5668918          Cross System Product/Application Query
------------------------------------------------------------------------------
 PF1=Help      2=          3=Quit          4=Return      5=Execute   6=
 PF7=          8=Forward   9=Sort(prodid)  10=Sort(desc) 11=         12=Cursor

====>
```

Figure 10-9. INSTFPP Panel 2

```
 SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 13 of 68
--------------------------------------------------------------------------------
          Type an X next to the product(s) you want to install
          When you have  finished,  press the PF5 key to begin
          the installation process.

_  5748XE4          Directory Maintenance
_  5748XXB          Display Management System for CMS
_  5664370          DisplayWrite/370
_  5748XX9          Document Composition Facility
_  5735XXB          Emulation Program
_  5654260          Environmental Recording Editing and Printing Program
_  5668890          Font Library Service Facility
_  5798DFH          FORTRAN Utilities
_  5668801          Graphical Data Display Manager - IMD
_  5668812          Graphical Data Display Manager - PGF
_  5668812    NL    Graphical Data Display Manager - PGF NL Feature
_  5664200          Graphical Data Display Manager Base
--------------------------------------------------------------------------------
 PF1=Help       2=           3=Quit           4=Return       5=Execute    6=
 PF7=Backward   8=Forward    9=Sort(prodid) 10=Sort(desc) 11=          12=Cursor

====>
```

Figure 10-10. INSTFPP Panel 3

```
 SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 25 of 68
--------------------------------------------------------------------------------
          Type an X next to the product(s) you want to install
          When you have  finished,  press the PF5 key to begin
          the installation process.

_  5664200    NL    Graphical Data Display Manager Base NL Feature
_  5668905          Graphical Display And Query Facility
_  5799AXX          Graphics Attachment Support Programming
_  5799BKE          Host Displaywriter Document Interchange
_  5668996          IBM BASIC/VM
_  5664185          IBM High-Accuracy Arithmetic Subroutine Library
_  5798DTE          IBM 3812 Pageprinter VM Support
_  5668897          Info Center/1
_  5668012          Interactive Instructional Presentation System
_  5664282          Interactive System Productivity Facility
_  5664285          Interactive System Productivity Facility/PDF
_  5664204    01    NetView Volume 1
--------------------------------------------------------------------------------
 PF1=Help       2=           3=Quit           4=Return       5=Execute    6=
 PF7=Backward   8=Forward    9=Sort(prodid) 10=Sort(desc) 11=          12=Cursor

====>
```

Figure 10-11. INSTFPP Panel 4

```
SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 37 of 68
------------------------------------------------------------------------------
         Type an X next to the product(s) you want to install
         When you have  finished,  press the PF5 key to begin
         the installation process.

_  5664204    02     NetView Volume 2
_  5664204    03     NetView Volume 3
_  5734PL3           OS PL/1 Optimizing Compiler and Libraries
_  5734LM5           OS PL/1 Transient Library
_  5664293           Overlay Generation Language
_  5664199           Page Printer Formatting Aid
_  5796PNQ           Pascal/VS
_  5664298           PC Bond
_  5664312           Print Services Access Facility
_  5664198    B      Print Services Facility/VM Base
_  5664198    S      Print Services Facility/VM 3800
_  5664198    V      Print Services Facility/VM 3820
------------------------------------------------------------------------------
 PF1=Help       2=           3=Quit          4=Return      5=Execute    6=
 PF7=Backward   8=Forward    9=Sort(prodid) 10=Sort(desc) 11=          12=Cursor

====>
```

Figure 10-12. INSTFPP Panel 5

```
SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 49 of 68
------------------------------------------------------------------------------
         Type an X next to the product(s) you want to install
         When you have  finished,  press the PF5 key to begin
         the installation process.

_  5664309           Professional Office System
_  5664309    PA     Professional Office System Applications Support Feature
_  5668AAA           Query Management Facility
_  5748XP1           RSCS Networking, Version 1
_  5664188           RSCS Networking, Version 2
_  5748XXJ           Structured Query Language/Data System
_  5664191           Virtual Machine Monitor Analysis Program
_  5664319           Virtual Machine/Personal Computer Host Server
_  5746AM2           Virtual Storage Extended/Virtual Storage Access Method
_  5664318           VM/Interactive Productivity Facility
_  5664364           VM Batch Facility
_  5798DMY           VM File Storage Facility
------------------------------------------------------------------------------
 PF1=Help       2=           3=Quit          4=Return      5=Execute    6=
 PF7=Backward   8=Forward    9=Sort(prodid) 10=Sort(desc) 11=          12=Cursor

====>
```

Figure 10-13. INSTFPP Panel 6

```
 SOINS02            INSTFPP PRODUCT SELECTION PANEL            Line 61 of 68
------------------------------------------------------------------------------
          Type an X next to the product(s) you want to install
          When you have  finished,  press the PF5 key to begin
          the installation process.

_  5748RC1          VM/Pass-through Facility
_  5796PNA          VM Real Time Monitor
_  5664283          VM/Integrated System-Productivity Facility
_  5664291          VMBACKUP Management System
_  5664292          VMTAPE Management System
_  5668958          VS COBOL II
_  5668806          VS FORTRAN
_  5664281          3270 PC File Transfer




------------------------------------------------------------------------------
 PF1=Help      2=          3=Quit          4=Return      5=Execute   6=
 PF7=Backward  8=          9=Sort(prodid) 10=Sort(desc) 11=         12=Cursor

====>
```

Figure 10-14. INSTFPP Panel 7

Part 3. VM/SP Service

Chapter 11. Introduction to VM/SP Service

This chapter describes general service concepts for VM/SP. You should be familiar with these concepts before you attempt to service any of the VM/SP components installed on your system.

When you have a servicing task, it is important for you to be able to distinguish between the types of code you are working with. For example, some components of VM/SP are maintained as source code. For source code, you update source statements in an assembler file and assemble that file to create a new executable module. Some components are maintained as object code. For object code, you replace preassembled executable modules or TEXT decks with new ones. Other components contain both source and object code. For these components, you may have to apply updates and replace TEXT decks.

In each case, the result of service is the same; you have an updated version of the code. However, there are significant differences in the processes you must go through to make the code executable.

In order to preserve the integrity of VM/SP, you should:

- Keep local user modifications on a separate minidisk from IBM supplied fixes.
- Never overlay the original base installation source and object files with the updated service versions.
- Backup the current service level before applying any other service.
- Apply service in a test environment before you install it on your production system.

Types of Service

In addition to the concept of source and object code, you need to know there are different types of service. They are:

Corrective Service
Preventive Service

Other IBM supplied fixes (ZAPs, over the phone updates)
Local updates[1]

Corrective Service

If you have a problem, IBM sends you (at your request) tapes with changes you can incorporate into the code of a product. These tapes contain both source updates and replacement parts that you request, to correct a problem you are having with a product. These changes are referred to as corrective fixes.

For corrective service to source code, refer to "Apply Corrective Service to Source Code" on page 12-8 for details. For corrective service to object code, refer to"Apply Corrective Service to CMS Object, GCS, and TSAF" on page 13-1 , or "Apply Corrective Service to Object Code" on page N-18 (for System Network Architecture products) for details.

Preventive Service

Periodically, IBM creates tapes containing replacement files for a product. The replacement files have changes already incorporated in them. Tapes containing this service are called preventive service tapes and are cumulative. For IBM products, these tapes are known as Program Update Tapes (PUTs).

For preventive service, you can use either the product's specific service exec or the VMSERV EXEC. For preventive service contained on a PUT, use the VMSERV EXEC. VMSERV, described on page 15-91, is an EXEC procedure included on the PUT that calls each product's service exec. The PUT DOCUMENT describes this procedure in detail.

VMSERV EXEC Functions

VMSERV:

- Controls positioning of the tape.
- Verifies the service level being applied (maps the PUT to create a service diskmap file).
- Prints the PUT DOCUMENT and Memo(s)-to-Users (optional).
- Issues each product's service EXEC to load service.
- Restarts to continue loading service
- Issues the service exec with the option to build.

---

[1] Throughout this chapter we use the words **fix** and **update** interchangeably.

Other IBM Supplied Fixes (ZAPs, over the phone updates)

This type of service usually involves direct over the phone assistance from the IBM support center. You may also need to apply updates to your system when other methods are not available. This involves creating your own control files, assigning minidisks for your fixes, and selecting the appropriate build procedure.

Refer to "Apply Local and IBM Corrective Updates to Source Code" on page 12-10 and to "Apply Emergency Fixes Using ZAPs" on page N-16 for details.

Local Updates

Local updates are user modifications to a component's source code. The procedures that you can use to apply local updates are similar to source updating CP and CMS. The difference is that you must create your own control files and auxiliary control files for applying a user modification. Refer to "Apply Local and IBM Corrective Updates to Source Code" on page 12-10 and to "Object Code Service Processing" on page N-16 for details.

A Virtual Machine for Updating VM/SP

In an attempt to standardize the procedures for service, we use the MAINT virtual machine in our examples for applying service to the VM/SP system. We strongly recommend you use this virtual machine when you update your system.

The MAINT virtual machine is initially assigned more privileges than other virtual machines. Generally, MAINT is given privilege classes ABCDEFG, which allows this virtual machine to issue any command in the system. Use caution when issuing commands (such as SHUTDOWN) from the MAINT userid.

MAINT has MDISK statements in its directory entry for all the system-related minidisks. Figure 11-1 on page 11-4 is a sample directory entry for the MAINT virtual machine. Figure 11-2 on page 11-5 shows MAINT's minidisks used for service.

```
USER MAINT CPCMS 16M 16M ABCDEFG
 ACCOUNT 1 SYSPROG
 OPTION ECMODE DIAG98
 IPL 190
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 123 3380 000 885 VMSRES MW RSYSRES  WSYSRES  MSYSRES
 MDISK 124 3380 000 885 VMPK01 MW RSYSRES  WSYSRES  MSYSRES
 MDISK 127 3380 000 885 VMPK04 MW RSYSRES  WSYSRES  MSYSRES
 MDISK 129 3380 000 885 PROFPK MW RSYSRES  WSYSRES  MSYSRES
 MDISK 130 3380 000 885 SQLPK  MW RSYSRES  WSYSRES  MSYSRES
 MDISK 19D 3380 229 048 VMPK01 MW ALL      WMAINT   MMAINT
 MDISK 190 3380 502 037 VMSRES MW ALL      WMAINT   MMAINT
 MDISK 191 3380 144 010 VMSRES MW RMAINT   WMAINT   MMAINT
 MDISK 193 3380 117 027 VMSRES MW RMAINT   WMAINT   MMAINT
 MDISK 194 3380 044 027 VMSRES MW RMAINT   WMAINT   MMAINT
 MDISK 196 3380 028 016 VMSRES MW RMAINT   WMAINT   MMAINT
 MDISK 201 3380 767 023 VMSRES MW RMAINT   WMAINT   MMAINT
 MDISK 293 3380 790 027 VMSRES MW RCMSAUX  WCMSAUX  MCMSAUX
 MDISK 294 3380 862 021 VMSRES MW RCPAUX   WCPAUX   MCPAUX
 MDISK 295 3380 211 014 VMSRES MW RUSRMOD  WUSRMOD  MUSRMOD
 MDISK 296 3380 070 019 VMPK01 MW RCPAUX   WCPAUX   MCPAUX
 MDISK 319 3380 021 006 VMSRES MW ALL      WMAINT   MMAINT
 MDISK 393 3380 353 063 VMPK04 WR RMAINT   WMAINT
 MDISK 394 3380 416 076 VMPK04 WR RMAINT   WMAINT
 MDISK 396 3380 499 034 VMPK04 WR RMAINT   WMAINT
 MDISK 492 3380 664 011 VMPK01 MW RTSFOBJ  WTSFOBJ  MTSFOBJ
 MDISK 494 3380 864 011 VMPK01 MW RTSFAUX  WTSFAUX  MTSFAUX
 MDISK 496 3380 092 001 VMPK01 MW RIPCX    WIPCSX   MIPSX
 MDISK 497 3380 492 007 VMPK04 MW RMAINT   WMAINT
 MDISK 59E 3380 875 010 VMPK01 MW ALL      WMAINT   MMAINT
 MDISK 595 3380 682 031 VMPK01 MW RMAINT   WMAINT   MMAINT
 MDISK 596 3380 713 021 VMPK01 MW RGCSAUX  WGCSAUX  MGCSAUX
*
```

Figure 11-1. Suggested VM Configuration for Updating a 3380 System

Refer to Appendix A, "Minidisks Reserved for the MAINT Userid" on page A-1 for the default contents of the minidisks listed in MAINT's directory entry.

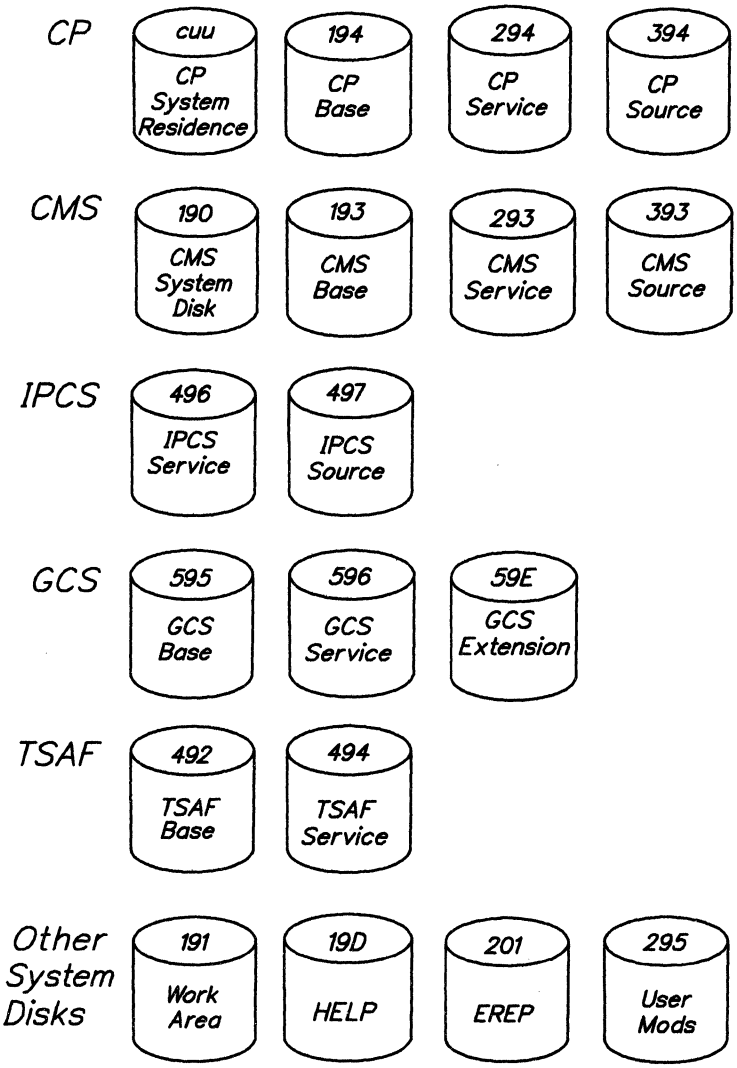


Figure 11-2. MAINT's Minidisks Used for Service

Minidisks Used for Service

The following describes the disks used during service processing.

Disk	Contents
Service	This disk contains executable system files that are loaded from the PUT.
Source	This disk should contain the source version of the base installation code.
System	This is the CMS system disk containing commands, EXECs, MACLIBs, and TXTLIBs.

Disk	Contents
Usermods	This disk contains any modifications you have made to the system.
Base	This disk contains the object version of the base installation code.

Accessing Disks

When you are using the VM/SP procedures to apply updates to system modules, be sure that you have the correct disks accessed in the appropriate search order.

Having the appropriate search order ensures that if a control file or an auxiliary control file with the same filename exists on two or more minidisks, the file located on the minidisk you accessed first in your search order will be the file used.

You may find it convenient to create EXEC procedures that do the links and accesses necessary for a particular update. For example, to update a CP module from files located on MAINT's 295, 294, 194, and 394 minidisks, your EXEC procedure might contain the following:

```
ACCESS 295 A  (Local modifications)
ACCESS 294 B/A (CP AUX files and updates)
ACCESS 194 C/A (CP TEXT files and MACLIBs)
ACCESS 394 D/A (CP source, COPY, and MACRO files)
```

Service Exec Minidisk Defaults

The service exec[2] uses the following minidisks as defaults when loading service. Service is loaded to the service disk for each component. Base installation code and original source code are never overlaid; **but, current service on the service disk is overlaid**. We recommend that you make a backup copy of your current service level before loading files from the PUT.

Note: If you want to use the SETUP function of the SPGEN EXEC for assemblies or module builds, you must tailor the SPGEN PROFILE. Refer to the disk addresses on the next page for the correct access orders. For information about the contents of SPGEN PROFILE, see "SPGEN PROFILE Syntax" on page 15-26, "SPGEN PROFILE Defaults" on page 15-29, and "Sample SPGEN PROFILE" on page C-3.

---

2 The service exec for VM/SP is 5664167D EXEC. Refer to the Memo-to-Users for details about the service exec processing.

Disk Addresses for Loading Service

The following are the service exec minidisk address defaults for loading service.

CP	CMS	IPCS	GCS	TSAF
294-service	293-service	496-service	596-service	494-service
294-source	293-source	496-source	496-IPCS service	496-IPCS service
	190-system			
	496-IPCS service			
	19D-HELP			

Note: The source disk option is allowed for new source files. The default for source files is the service disk.

Disk Addresses for Updating

The following are the service exec minidisk address defaults and search order for applying updates.

CP	CMS	IPCS	GCS	TSAF
295-usermods[3]	295-usermods[3]	295-usermods[3]	295-usermods[3]	295-usermods[3]
294-service	293-service	496-service	596-service	494-service
194-base	193-base	497-source	595-base	492-base
394-source	393-source	294-CP service[4]	293-CMS service	293-CMS service
293-CMS service	294-CP service[4]	194-CP base[4]	193-CMS base	193-CMS base
193-CMS base	194-CP base[4]	293-CMS service		
		193-CMS base		

Disk Addresses for Building

The following are the service exec minidisk address defaults and search order for building:

CP	CMS	IPCS	GCS	TSAF
295-usermods[3]	295-usermods[3]	295-usermods[3]	295-usermods[3]	295-usermods[3]
294-service	293-service	496-service	596-service	494-service
194-base	193-CMS base	293-CMS service	595-base	492-base
293-CMS service		193-CMS base	293-CMS service	293-CMS service
193-CMS base			193-CMS base	193-CMS base

---

[3] User modifications. These are any changes you create to put on your system.

[4] Some CMS and IPCS modules require access to the latest version of the CP macro library (DMKSP MACLIB) for reassembly.

Notes:

1. *If you have created any user modifications, you must access the minidisk(s) containing those changes first in your access order.*

2. *You access the CMS base and service minidisks for service utilities and the SPGEN EXEC.*

Disk Access Profiles

In addition to the service exec minidisk address defaults, there are two different disk access profiles related to servicing. The service process you use determines which profile is used.

VMPUT PRODUCTS Inventory File

This file, which you create, has an entry for each product that is to be serviced from the PUT. VMSERV uses this file when loading service and during build processing. The purpose of this file is to automatically select which products will have service applied from the PUT and to pass specific parameters to each products' service exec. If this file exists, VMSERV scans it and applies service to the products that have a complete entry in this file.

If the entry is not complete or the file does not exist, then the service exec default addresses (shown on page 11-7) are used.

A complete entry in the VMPUT PRODUCTS Inventory file is the program number followed by 17 positional disk parameters. The format of an entry is:

```
5664167D (p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13
          p14 p15 p16 p17)
```

where:

p1 CP base minidisk (TEXT, MACLIB, etc.)
p2 CP service minidisk (AUX and update files)
p3 CP source minidisk (ASSEMBLE files)
p4 CMS system disk
p5 CMS base minidisk
p6 CMS - IPCS service minidisk
p7 CMS service minidisk
p8 CMS source files
p9 HELP files
p10 IOCP I/O configuration program for 3081
p11 IPCS service minidisk and interface files
p12 IPCS source minidisk
p13 GCS - IPCS service minidisk
p14 GCS base minidisk
p15 GCS service minidisk

p16 TSAF - IPCS service minidisk
p17 TSAF service minidisk

If you invoke the service exec without disk parameters, the service exec minidisk defaults are used.

SPGEN PROFILE File

This file contains information that the system generation tool (SPGEN EXEC) uses when building system nuclei. This profile contains disk addresses that you may specify to be used for a nucleus build process. If you use VMSERV or the service exec to handle the building of CP, CMS, or GCS, the disk accesses in the SPGEN profile are ignored because the service exec passes the NOSETUP option to SPGEN. The ones identified during execution of the service exec are used (either from the VMPUT PRODUCTS file, or from specifying addresses at a service exec prompt or accepting the service exec defaults). If you invoke SPGEN directly, the SPGEN PROFILE disk accesses are used. (You can also access the disks in the order you need and invoke SPGEN with the NOSETUP option).

For information about the contents of SPGEN PROFILE, see "SPGEN PROFILE Syntax" on page 15-26 and "SPGEN PROFILE Defaults" on page 15-29.

Module Identifiers

Each of the components of VM/SP has a unique character module identifier, which is used to name the component's modules. These module identifiers are also used to name the files that are used to update the components. The identifiers are:

Component	Module Identifier
CP	DMK
CMS	DMS
IPCS	DMM
GCS	CSI
TSAF	ATS

The default CMS filetypes are used to identify the source code, object code, module files, and libraries associated with each component. These filetypes are:

Filetypes	Type of File
ASSEMBLE	Source File
MODULE	Nonrelocatable object code
MACLIB	Macro or copy library
TEXT	Object code (relocatable)
TXTAP	Object code with attached processor support (relocatable)
TXTAPFT	Object code with attached processor (with CP FRET Trap)
TXTLIB	Text file library
TXTMP	Object code with multiprocessor support (relocatable)
TXTMPFT	Object code with multiprocessor support (with CP FRET Trap)
TXTlangid	Object code for language files (langid is the identifier for a language)
TXTUP	Object code with uniprocessor support
TXTUPFT	Object code with uniprocessor support (with CP FRET Trap)

Programs for Servicing VM/SP

Figure 11-3 below shows the programs available for updating VM/SP.

Program	Comments
ASMGEND	Updates the VM/370 system assembler using GENMOD.
CMSGEND	Creates a new CMS command module from updated TEXT files using GENMOD.
DCSSGEN	Rebuilds the CMSINST discontiguous saved segment from updated TEXT files.
DOSGEN	Rebuilds the CMSDOS and INSTVSAM discontiguous saved segments from updated TEXT files.
GENMOD	Creates a nonrelocatable module file on a CMS disk.
GENTSAF	Rebuilds the TSAF load module from updated text files using a loadlist (e.g. ATSLOAD EXEC) and an optional control file as input.
PRELOAD	Collects multiple text files and reformats them into a single text file.
SAMGEN	Rebuilds the CMSBAM discontiguous saved segments from updated TEXT files.
SPGEN	Does various system generation and maintenance functions, using the parameters contained in SPGEN PROFILE.
VMFASM	Creates an updated source file using IBM updates, PTFs, and user updates, then assembles the source file.
VMFLKED	Invokes the CMS LKED command to link-edit modules into a LOADLIB
VMFLOAD	Creates a new CP or CMS nucleus based on a control file and a loadlist EXEC file.
VMFMAC	Creates macro libraries using IBM and user updates.
VMFNLS	Applies updates to national language files and compiles the updated versions.
VMFTXT	Creates text libraries using IBM updates.
VMSERV	Controls the individual service EXECs on the system Program Update Tape (PUT).
VRSIZE	Generates DMKSLC text which is used to generate a V=R area in the system.
VSAMGEN	Rebuilds the CMSVSAM and CMSAMS discontiguous saved segments from updated TEXT files.
VSEVSAM	Builds a VSE/VSAM MACLIB containing the supported VSE/VSAM macros.

Figure 11-3 (Part 1 of 2). Programs for Servicing VM/SP

Program	Comments
ZAP	Modifies or dumps MODULE, LOADLIB, or TXTLIB files.
ZAPTEXT	Modifies or dumps individual text files.

Figure 11-3 (Part 2 of 2). Programs for Servicing VM/SP

Other service procedures, such as VMFMERGE EXEC, VMFREMOV EXEC, and VMFZAP EXEC, are available to service Systems Network Architecture (SNA) products. See Appendix N, "Servicing Systems Network Architecture (SNA) Products" on page N-1 for examples of how to use those procedures.

Chapter 15, "EXEC and Command Procedures" contains information about the SNA service programs and the programs listed in Figure 11-3.

Make sure that you understand the concepts in this chapter before you continue.

Chapter 12. Servicing Source Maintained VM/SP Products

This chapter contains information about how to service products that are maintained by updating source statements and reassembling them into object files. In this chapter we discuss several execs.

VMFMAC creates macro libraries using IBM and user updates.
VMFASM updates a source file using IBM fixes and PTFs and user updates, then assembles the updated source file.
VMFTXT creates text libraries.
VMFNLS applies updates to national language files and compiles the updated versions.
VMFLOAD creates a new CP or CMS nucleus based on a control file and a loadlist EXEC file.

Note that the procedures outlined in Figure 12-1 show how to update source code for VM/SP. Not all the available service programs are mentioned.

Deciding Which Service Procedure to Use for Updating Source

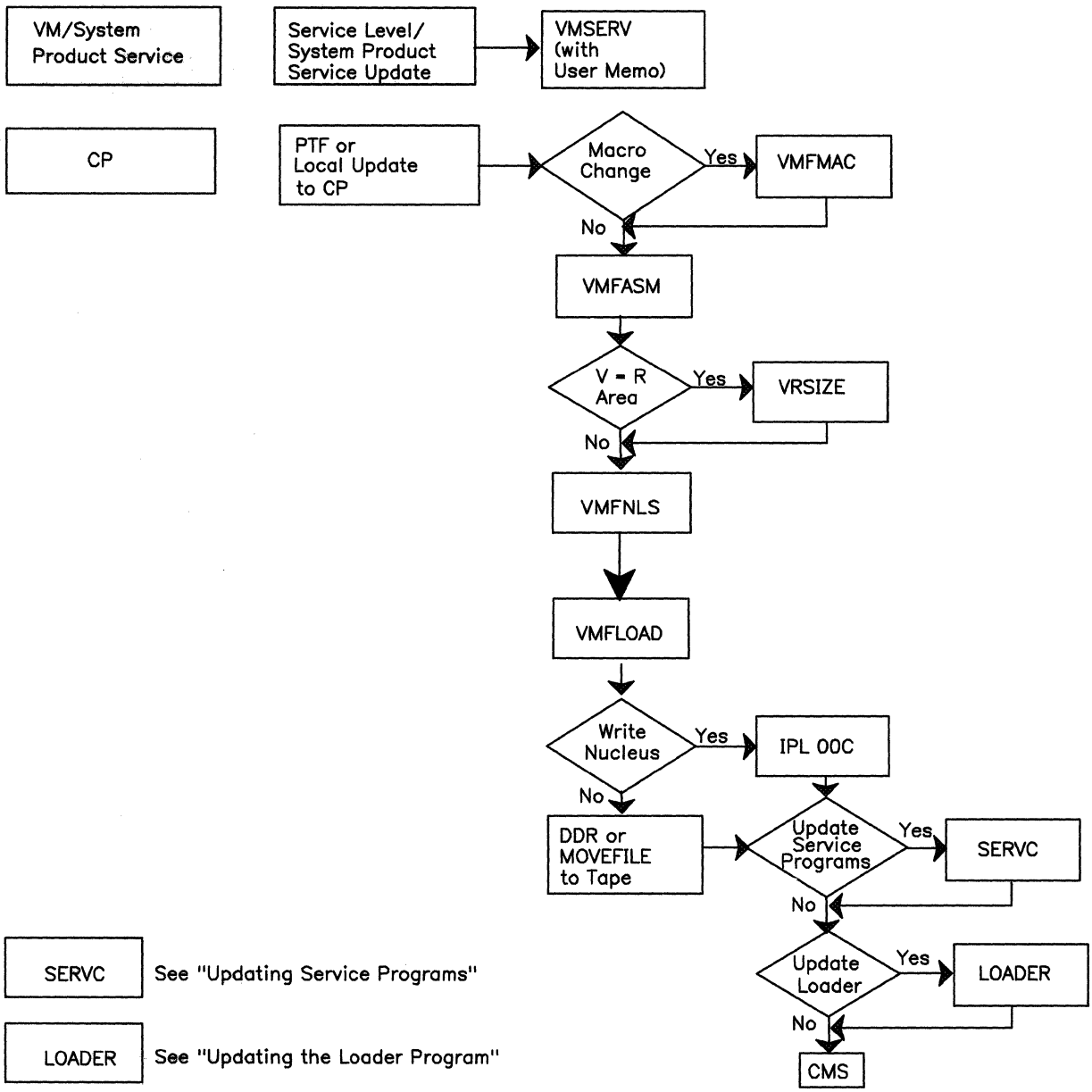


Figure 12-1 (Part 1 of 2). Deciding Which Updating Procedures to Use

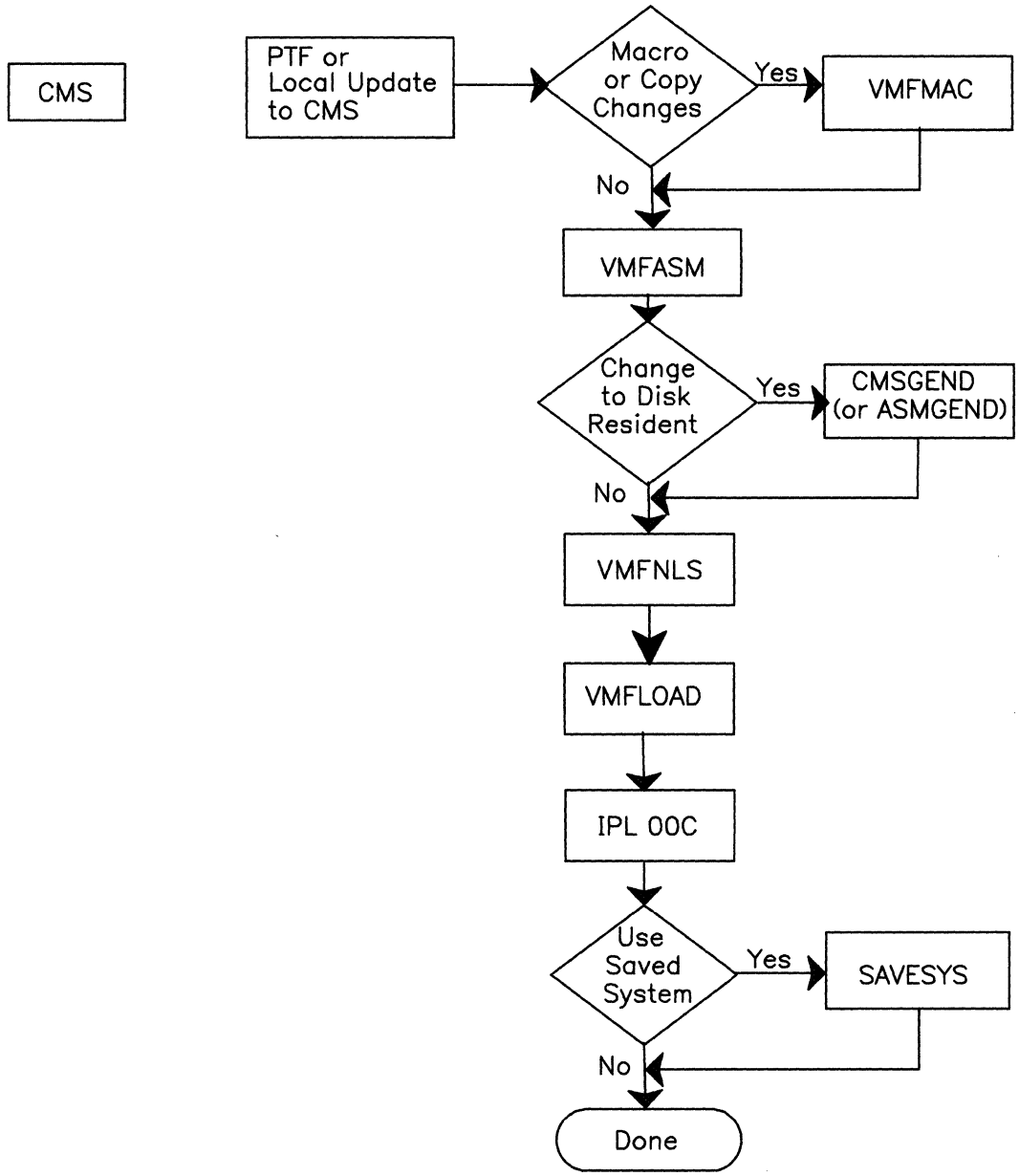


Figure 12-1 (Part 2 of 2). Deciding Which Updating Procedures to Use

Before you use any of the procedures, you need to establish a virtual machine userid for maintenance tasks. The procedures we describe use the MAINT virtual machine. For more information about the use of this virtual machine for servicing VM/SP, refer to "A Virtual Machine for Updating VM/SP" on page 11-3.

You must also be very familiar with the CMS files that are used for updating and the naming conventions used by IBM.

Files for Updating Source

The VM/SP service process relies heavily on CMS files called control files. There are several of these control files in the VM/SP system depending on the configuration.

There are control files, auxiliary control files, and the actual update files. These files may have the following generic filetypes:

Filetype	File Contents
CNTRL	Control file
AUXxxxxx	Auxiliary control file
UPDTxxxx	Local update (listed in a CNTRL file)
anything	Local update (listed in an AUX file)

Note: An AUXSP file may have the release number appended to it as the last two-character component identifier. Control files may be extremely simple or quite elaborate, but their important function is to allow you to tailor your system to meet your current needs.

Control Files

Control files are used in:

- updating source files
- assembling source code using the VMFASM EXEC procedure
- building MACLIBs using the VMFMAC EXEC procedure
- creating TXTLIBs using the VMFTXT EXEC procedure
- regenerating updated EXECs.
- creating nucleus load decks

Control files are used by the CMS UPDATE command. Both the VMFMAC and VMFASM update procedures invoke UPDATE with the CTL option to modify source files. For VMFMAC and VMFASM, the control file must have a filetype of CNTRL. The VMFLOAD and VMFTXT programs also use a control file; this is usually the same control file used by the VMFASM EXEC.

Sample Control File

For an understanding of how the update procedures work, you should be familiar with the elements in a control file. A typical control file supplied with the VM/SP system may look like Figure 12-2.

```
*THIS IS A SAMPLE CNTRL FILE FOR LOCAL CP UPDATES
TEXT MACS LOCALIB DMKSP DMKMAC DMSSP CMSLIB OSMACRO
UP   UPDTFIX1
LOC1 AUXLCL
TEXT AUXSP
```

Figure 12-2. Sample Control File

Notes

- An asterisk (*) in the first column indicates a comment record.
- The first field in each of the remaining lines is the update level identifier. You can think of this as a name field. VMFASM, VMFTXT, and VMFLOAD use the update level identifier to identify the text deck.

  - VMFASM uses the update level identifier of the most recent update that was found and applied as the filetype of the text deck produced by the assembly.
  - VMFTXT uses the update level identifiers to locate text decks when building a TXTLIB.
  - VMFLOAD uses the update level identifiers to locate text decks when punching a new CP or CMS nucleus.

  The update level identifier on the MACS record (TEXT) is used by VMFASM to name an assembled update text deck when no update files are found; it is also used by VMFLOAD and VMFTXT when it fails to locate a text file based on update level identifiers associated with update files or auxiliary control files.

- The second field in each of the remaining lines defines the statement's function.

  VMFASM uses the library list from the MACS record to issue a GLOBAL command before assembling the updated source file. The libraries are searched in the order specified. DMKSPA should precede DMKSP if AP support is required. If MP support is required DMKSPM should precede DMKSP.

  The characters UPDT identify the filetype of a single update file, UPDTFIX1 in this example. You can omit the characters UPDT, in which case the record would read UP FIX1 (because UPDT is assumed).

  The characters AUX identify an auxiliary control file that lists additional fixes to be applied; AUXLCL in this example.

AUXSP is the VM/SP auxiliary control file, listing updates distributed by IBM. This file is listed at the bottom of the control file so that these updates are applied first. (Note, the file is read from the bottom to the top).

A control file can have any number of update identification (UPDTxxxx) records, AUX file identification (AUXxxxxx) records, and comments, but can have only one MACS record.

Refer to Appendix L, "Control File Identifiers" on page L-1 to determine which CNTRL file represents your system configuration.

Auxiliary Control Files

All auxiliary control files distributed by IBM have the filetype AUXSP (AUXMSP for CMS MACLIB updates). When an update is distributed for a module, an auxiliary control file is also distributed.

All of the PTF and update files distributed by VM/SP are assigned filetypes as follows:

Onnnnnxx
Unnnnnxx

where

O indicates a System Product module update (or System Product Interpreter EXEC change).

U indicates a System Product CMS macro update.

nnnnn is an APAR or PTF number.

xx is the two-character component identifier (DK, DM, or DS for CP, IPCS, and CMS respectively).

For example, the code and fixes to answer APAR VM12567 against the CP module DMKCFM in the VM/System Product are contained in the file DMKCFM O12567DK. The file DMKCFM AUXSP contains the entry:

```
O12567DK -  COMMENT DESCRIBING FIX
```

When you create files for local corrective service or updates of VM/SP modules, you should create a local control file, that is, a copy of the appropriate VM/SP control file with an entry for your AUXLCL auxiliary control file and the filename of your local MACLIB (if appropriate). For example, the file CPLCL CNTRL may contain:

```
TEXT MACS YOURMAC DMKSP DMKMAC DMSSP CMSLIB OSMACRO
LCL AUXLCL
TEXT AUXSP
```

Note: Control files (CNTRL) and auxiliary control files (AUXxxxxx) are always read from the bottom up. The IBM-supplied auxiliary files should be the bottom entry in the control file so that the IBM updates are applied first.

Source Code Service Processing

There are many ways to apply service to components of your system that are source code maintained. This section describes, in a very generic way, how to apply:

- Corrective Service
- Local and IBM Corrective Updates
- Preventive Service

Apply Corrective Service to Source Code

Corrective service is the application of a Program Temporary Fix (PTF) or an IBM Change Team supplied fix to correct a problem.

Corrective fixes can be:

- *fixes on a PTF tape* that are a result of closed valid APARs. Either Program Support Services (PSS) or the Change Teams provide these fixes.

- *relief fixes* to fix severe problems provided by the Change Teams. These are fixes sent on a tape prior to APAR closure.

To apply corrective fixes, here's what you need to do.

1. Call you IBM support center for the latest service information for the type of service you are doing.

2. Create a good backup copy of your system.

3. Make sure you have the correct minidisk search order.

4. Mount and attach the service tape.

5. Use VMFPLC2 to load the first physical file on the corrective service tape. The first tape file contains information for you in the cover letter and in the Program Identification Number (PIN) pages.

6. Read all the service documentation sent to you with your service.

7. Use VMFPLC2 again to load the updates files on the tape. The service document explains the contents of the tape and how many files there are. If you do not want to load the entire tape, use the SCAN option of the VMFPLC2 command to find the particular file(s) you want to load. (Refer to "VMFPLC2 MODULE" on page 15-72 for more information about VMFPLC2). At this time you want to load:

   - The actual update files
   - The update file prerequisites.

8. Continue service processing as described in "Apply Local and IBM Corrective Updates to Source Code" on page 12-10.

Note: If you need to apply service to both CMS source and object code, also refer to "Apply Corrective Service to CMS Object, GCS, and TSAF" on page 13-1.

Apply Local and IBM Corrective Updates to Source Code

This section assumes you have added all updates to your AUX files.

Suppose that you want to update CP, and then build a new CP nucleus. You should keep in mind that the procedure for updating source files and macro libraries are the same for all the VM/SP components, and that the procedure for punching a new CMS nucleus is basically the same as the procedure for punching a new CP nucleus.

For local (user) fixes, you need to create an update file that contains UPDATE statements to change the source code. You can also create a control file listing the update or auxiliary files you want to update.

The 295 minidisk is defined to store your local, user-written fixes and control files. With your updates on this disk, you can service your code separately from IBM's. In the following example you are going to:

1. Update a user MACLIB. You want to change two members of DMKMAC MACLIB, placing the updated members in local MACLIB, YOURMAC MACLIB on MAINT's 295 minidisk. The members you want to change are:

   ACCTON COPY (for accounting routines)

   RDEVICE MACRO

   Changes to ACCTON COPY require you to reassemble DMKACO. Changes to RDEVICE MACRO require you to reassemble DMKRIO.

2. Add a CP command. If the command has already been assembled into the file DMKCMD TXTLCL on MAINT's 295 minidisk, then you must update the CP module DMKCFC to recognize the new command name.

We begin with updating the macro library, YOURMAC, using VMFMAC. Figure 12-3 on page 12-12 shows the files used by VMFMAC in this example. Next we use VMFASM EXEC to reassemble the modules that reference the changed macros. Figure 12-4 on page 12-15 shows the files used by VMFASM.

After you have reassembled all the modules that require updating you may build a new CP nucleus that contains the updated text decks. Figure 12-6 shows the files used by VMFLOAD.

Using VMFMAC to Update Macro Libraries

The VMFMAC EXEC procedure is specifically designed to update macro libraries. You *must* provide:

- An EXEC file listing the names of the members to be included in the macro library. The filename of the EXEC matches the filename of the MACLIB being created.

You can *optionally* provide:

- A control file that lists update files or auxiliary control files to be applied. In our example, the control file is called YOUROWN CNTRL and the auxiliary control files are:

  ```
  ACCTON  AUXSP
  RDEVICE AUXSP
  RDEVICE AUXLCL
  ```

- Update files, with UPDATE control statements to change the macro library members. VMFMAC calls the CMS UPDATE command to update the members. In our example these are:

  ```
  ACCTON  O12567DK
  ACCTON  O12263DK
  ACCTON  UDPTLCL
  RDEVICE FIXPTF
  RDEVICE O12024DK
  ```

  All of the updates files listed above, except ACCTON UDPTLCL, are referenced by the auxiliary control files. ACCTON UDPTLCL is referenced in the YOUROWN CNTRL control file.

  You must also have available any IBM PTFs that have been distributed for the macro library.

Refer to "VMFMAC EXEC" on page 15-59 for a description of the VMFMAC EXEC command format and more information about this exec.

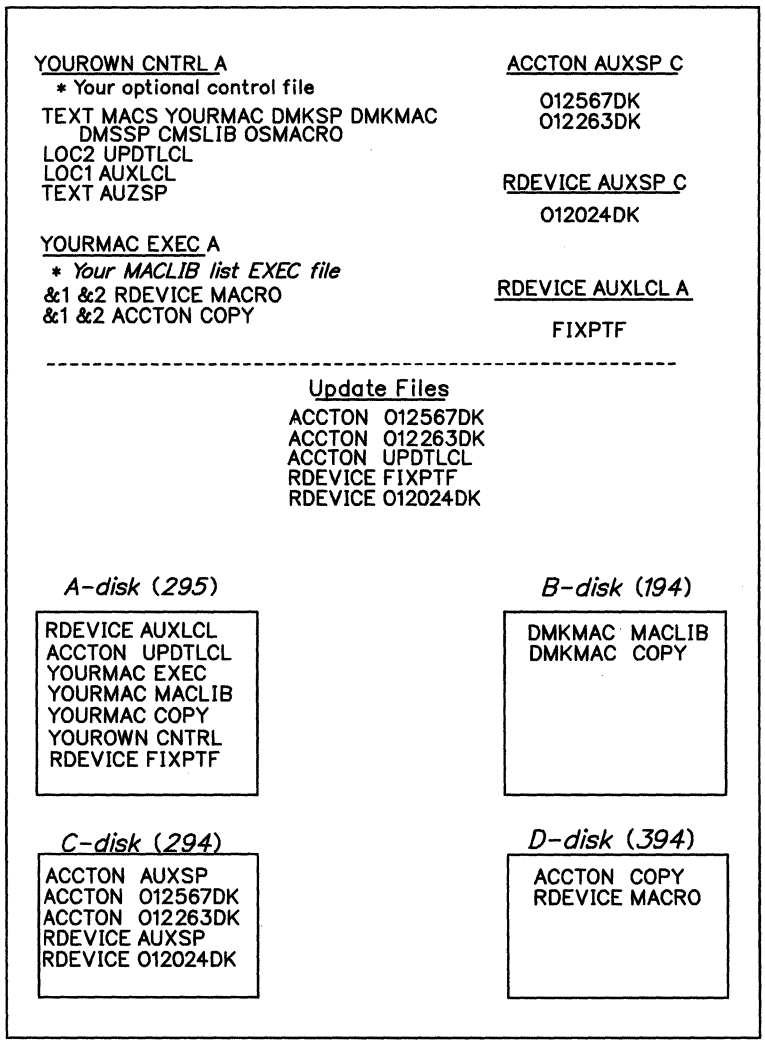


Figure 12-3. Files for VMFMAC

Figure 12-3 shows the files used in our example to update RDEVICE MACRO and ACCTON COPY. To incorporate these updates, you should have access to the source COPY and MACRO files on MAINT's 394, and the CMS system disk. The search order should be:

```
295 A    R/W
294 B/A  R/O
194 C/A  R/O
394 D/A  R/O
190 S    R/O
```

If no errors occur during this procedure, VMFMAC erases the existing MACLIB and writes a new one. VMFMAC also writes a new YOURMAC COPY file.

In the following example we use an EXEC file, we created, called YOURMAC EXEC. When you issue the command:

vmfmac yourmac yourown

the VMFMAC EXEC procedure uses the YOURMAC EXEC to rebuild YOURMAC MACLIB. When you specify a control file, VMFMAC applies any updates that are referenced. If no updates exist, or you did not specify a control file, then the existing MACRO or COPY file is included in the new MACLIB without changes. In this example, the YOURMAC EXEC contains RDEVICE MACRO and ACCTON COPY and each have updates.

- For ACCTON COPY, the UPDATE command looks for a file called ACCTON AUXSP first. This file exists and it contains O12567DK and O12263DK. Remember that processing is done from the bottom of the file to the top, so update files ACCTON O12263DK and ACCTON O12567DK are applied, in that order.

- Next UPDATE looks for a file called ACCTON AUXLCL. This file does not exist so no update is made.

- Finally UPDATE looks for a file called ACCTON UDPTLCL. This update file exists, so it is applied.

- For RDEVICE MACRO, UPDATE looks for a file called RDEVICE AUXSP first. This file exists and it contains O12024DK, so update RDEVICE O12024DK is applied.

- Next UPDATE looks for a file called RDEVICE AUXLCL. This file also exists and it contains FIXPTF, so update RDEVICE FIXPTF is applied.

- Finally UPDATE looks for a file called RDEVICE UPDTLCL. This file does not exist so no additional update is made.

After the entries in the YOURMAC EXEC are processed, VMFMAC erases the existing YOURMAC MACLIB and creates a new YOURMAC MACLIB with the updated members. An additional file, YOURMAC COPY, is produced; this file contains a record of the updates that were applied. YOURMAC COPY is also added to YOURMAC MACLIB to provide you with a record of changes.

Since the ACCTON COPY is modified, you have to reassemble DMKACO. In addition, changes to the RDEVICE macro require you to reassemble DMKRIO. You use the VMFASM EXEC for reassemblies.

Refer to "Using VMFASM to Update Source Files" on page 12-14 for this procedure.

Using VMFASM to Update Source Files

You use the VMFASM EXEC procedure to update a specific source file according to entries in a control file. To assemble the updated source you *must* provide:

- The filename of the source file to be updated. This file **must** have a filetype of ASSEMBLE (in the following examples, we use the ASSEMBLE files: DMKACO, DMKRIO, and DMKCFC.
- A control file that lists the update files and auxiliary control files (in the following example, we use the control file called YOUROWN CNTRL).

You can *optionally* provide:

- Any non-default options.

Refer to "VMFASM EXEC" on page 15-38 for a description of the VMFASM EXEC command format, options, and more information about this exec.

In the previous example, "Using VMFMAC to Update Macro Libraries" on page 12-11, you updated a MACLIB. You changed ACCTON COPY, so now you have to reassemble DMKACO. In addition, changes to the RDEVICE macro require you to reassemble DMKRIO. You should create dummy updates of the form:

```
File:  DMKRIO  FIXPTF
./* Indicates reassemble of DMKRIO for RDEVICE FIXPTF
```

If no files with filetype UPDTLCL exist, then the resulting text decks are named TXTLOC2. The filetype indicates which modules need to be reassembled for the local macro FIXPTF.

To reassemble DMKACO and DMKRIO, you use the new MACLIB and your own control file to update them. You do this by issuing the following commands:

```
vmfasm dmkaco yourown
vmfasm dmkrio yourown
```

You must be sure that all the current PTFs and auxiliary control files shipped with the base or PUT are available on MAINT's 294, and all local PTFs and auxiliary control files are available on MAINT's 295.

Remember to change your control file to include DMKSPA for an attached processor system, or DMKSPM for a multiprocessor system.

The second update we want to make is to add a CP command. The command has already been assembled into the file DMKCMS TXTLCL on MAINT's 295 minidisk. We must update the CP module DMKCFC to recognize the new command name. We do this using the VMFASM EXEC

as shown on the next few pages. Figure 12-4 on page 12-15 shows the files used for updating and assembling the module DMKCFC.

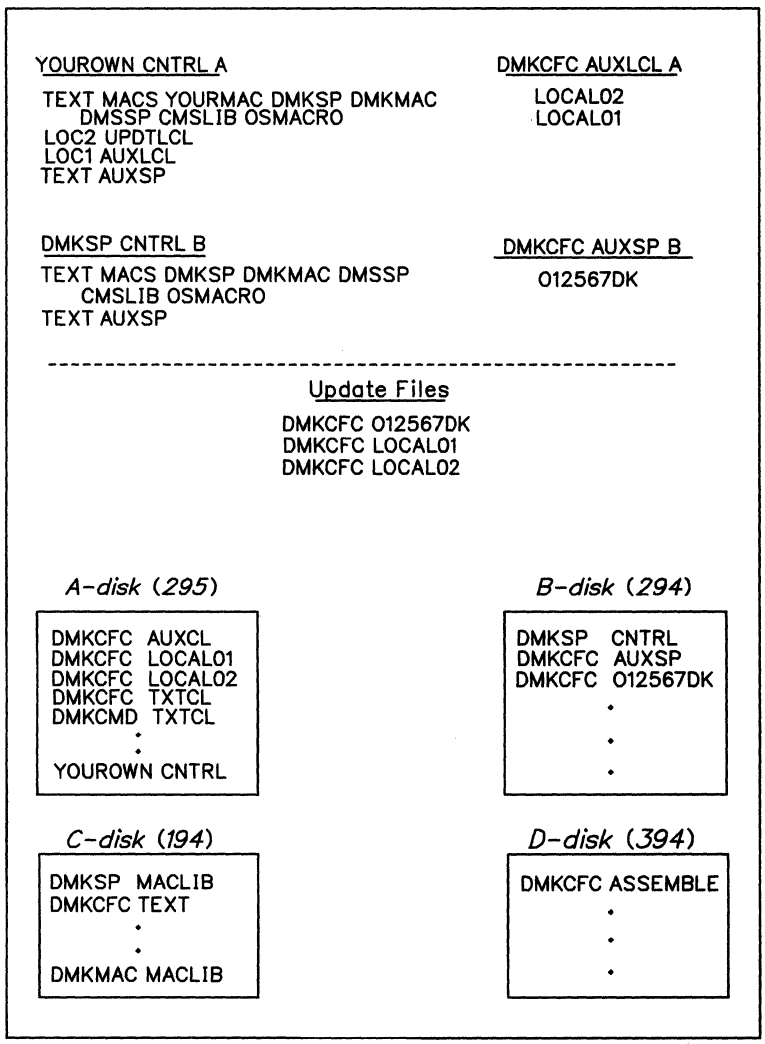


Figure 12-4. Files for VMFASM

In addition to the 294 and 295 minidisks, you should also have access to the 194, 394, and 190 minidisks. The search order is:

```
295 A    R/W
294 B/A  R/O
194 C/A  R/O
394 D/A  R/O
190 S    R/O
```

When you issue the command:

vmfasm dmkcfc yourown

the VMFASM EXEC procedure invokes the UPDATE command with the CTL, STK, PRINT, and OUTMODE A1 options.

Refer to "VMFASM EXEC" on page 15-38 for a description of the VMFASM EXEC command format, options, and more information about this exec.

- For DMKCFC AUXSP, UPDATE looks for the file DMKCFC O12567DK. This update file exists, so it is applied and processing continues with the next auxiliary control file.

- For DMKCFC AUXLCL, UPDATE looks for files DMKCFC LOCAL01 and DMKCFC LOCAL02. These files exist and are applied in that order (remember processing is done from the bottom of the file to the top).

- Since no file named DMKCFC UPDTLCL exists, no update is applied for that entry in the control file.

When all the updates have been applied, VMFASM issues a GLOBAL MACLIB command for the MACLIBs listed on the TEXT MACS statement in the YOUROWN CNTRL file.

When the GLOBAL command is complete, VMFASM calls the assembler to assemble the updated source file, which has a temporary filename of $DMKCFC. (If no updates are applied, the temporary filename is DMKCFC.)

After the assembly is complete, VMFASM uses the update level identifier of the most recent update that was found and applied by the UPDATE command to rename the TEXT file produced by the assembly. In our example, the output file is named DMKCFC TXTLOC1. This file is written on the 295 (A) disk that **must** be accessed Read/Write.

The temporary file created by the UPDATE command is erased and the original source file remains untouched on the 394 disk.

The UPDATES file produced by a multi-level update is concatenated with the output text deck (object code). This UPDATES file contains comments about the fixes that were applied and the order in which they were applied.

Notes:

1. *If there is not enough space available on your A-disk for the assemble work files, the following message appears:*

   DMSERD107S Disk A(0xxx) is full

   and the assembly ends.

   To correct the situation, erase any unneeded files from your A-disk or use a temporary R/W minidisk. Then restart the assemble process.

2. *To prevent you from inadvertently including a down-level textfile in your nucleus, VMFASM erases any TEXT filetypes on your A-disk for the module you are assembling. A new textfile name other than TEXT, such as TXTLOC1 or TXTLOC2, is erased. For a new textfile name of TEXT, the new file replaces the existing TEXT file on the A-disk.*

Using PSPACE to Determine Maximum Module Size

When you apply local updates to the source code, you may discover an assembly error generated by the PSPACE macro.

The PSPACE macro determines if a module has sufficient addressable space at its end for the addition of a patch area, if one is required for service. It insures that the module's existing base register(s) are sufficient to be used for addressability in the patch space. This macro uses 6 bytes of storage (+1 for alignment, if needed.)

Note: This macro does not actually generate a patch area. It only determines if there is enough room to add a patch area when it is needed.

The format of the PSPACE macro is:

[*csect*]	**PSPACE**	[**PERCENT=** {*nnn* | **5**} **END=** {*label* | *}]

where:

csect
is the name of the control section to be checked. If **csect** is not specified, the label on the START or most recent CSECT statement is used.

PERCENT
is the size of the minimum allowable patch area. It is expressed as a percentage of the current size of the module. The default is 5%.

END
specifies the label of an instruction on a halfword boundary that follows the last instruction of the CSECT. That is, **label** marks the end of the CSECT. If **END** is not specified, it is assumed that the PSPACE macro marks the end of the CSECT.

Placement of the Macro

For modules with only one entry point, the macro may be placed after the last statement in the desired CSECT. To insure proper placement, code the *csect*, and place it immediately before the END statement. If the macro is not placed in a CSECT, the following error may appear for the statement generated by the macro:

```
IFO217 RELOCATABILITY ERROR NEAR OPERAND COLUMN nn
```

An alternative is to place the macro after the last executable instruction in the CSECT, and use the END= operand to define the end of the CSECT (module).

For modules containing multiple entry points that establish a common addressability (to the beginning of the module, for instance), you can follow

the procedure above. If a module's entry points establish their own addressability, the PSPACE macro should be placed after the last executable instruction in the *first* entry point. You must use the END= operand to identify the end of the CSECT.

If the requested amount of patch area space does not exist, the following assembly error appears for the statement generated by the macro:

```
IF0209 ADDRESSABILITY ERROR- BASE AND DISPLACEMENT CANNOT BE
       RESOLVED AND ARE SET TO 0.
```

If you wish to force assembly, code PERCENT=0 and reassemble the module.

Using VMFTXT to Create Text Libraries

Use the VMFTXT EXEC procedure to create text libraries. Before you use VMFTXT, you *must* provide:

- An EXEC file listing the names of the members to be included in the text library (in the following example we use an EXEC file called YOURMEMB EXEC).

You can *optionally* provide:

- A control file that lists update files or auxiliary control files (in the following example we use a control file called YOUROWN CNTRL). The update files and auxiliary files identify the filetypes to be searched for the library names.

Refer to "VMFTXT EXEC" on page 15-84 for a description of the VMFTXT EXEC command format and more information about this exec.

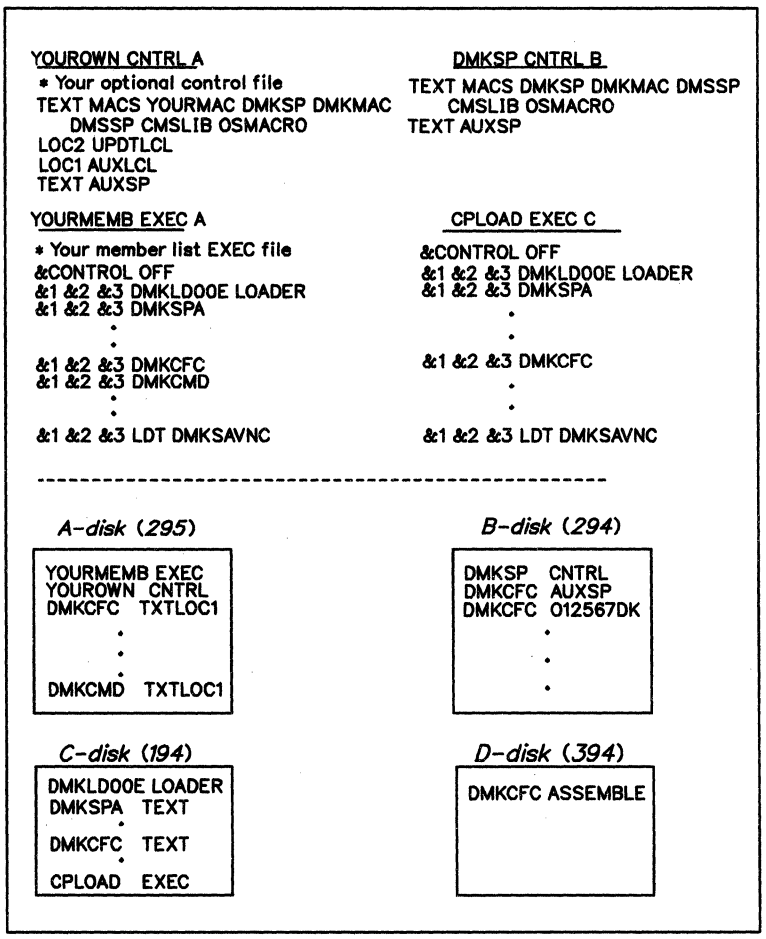


Figure 12-5. Files for VMFTXT

Figure 12-5 shows the files used for creating a new YOURMEMB TXTLIB. This TXTLIB incorporates the updates described on the previous pages. The search order should be:

```
295 A   R/W
294 B/A R/O
194 C/A R/O
394 D/A R/O
190 S   R/O
```

The following is an example of using VMFTXT specifying an optional control file. When you issue the command:

vmftxt yourmemb [yourown]

the VMFTXT EXEC procedure uses the YOURMEMB EXEC to determine which members to include. In this example, YOURMEMB EXEC is identical to the distributed CPLOAD EXEC file, except there is an added entry for your DMKCMD file.

- VMFTXT takes the files from the member list (YOURMEMB EXEC) and adds them to the library. They are added in the order they appear in the member list. Thus, in this example, VMFTXT adds DMKLD00E LOADER first.

  If you specify a filetype in the member list (as is the case with DMKLD00E LOADER), then VMFTXT adds that specific file. Normally you do not specify a filetype, in which case VMFTXT uses the update level identifier in the control file (in our example, the identifiers are TEXT, LOC1, and LOC2) to determine the filetype.

- DMKSPA, the next entry in the member list, does not have a filetype, thus VMFTXT looks at the control file. Since control files are structured so that the most recent update is at the top of the file, VMFTXT begins reading at the top of the control file. In the example, the update level identifier for the first update record is LOC2.

- VMFTXT searches for the file DMKSPA TXTLOC2. This file does not exist on any of the accessed disks.

- VMFTXT looks at the next identifier, LOC1, and searches for DMKSPA TXTLOC1. This file does not exist on any accessed disks either.

- VMFTXT reads the next identifier, TEXT and searches for DMKSPA TEXT. This file exists on the 194 disk, so VMFTXT adds it to the library you want to build.

- Then VMFTXT looks at the next entry in the member list and repeats the entire process.

  When VMFTXT reaches the entry for DMKCFC in the member list, it locates the file DMKCFC TXTLOC1 on the 295 disk. The DMKCFC module contains your updates. (Note that VMFTXT takes the updates

from the 295 disk and not from the DMKCFC TEXT on the 194 disk. This is because we have 295 accessed before 194 in the search order).

The process continues in this way until VMFTXT reaches the end of the member list EXEC file. For each member VMFTXT adds, you receive the message:

```
DMSWTX895I Member fn ft added.
```

When VMFTXT adds all the members, you receive the message:

```
DMSWTX895I Member VMFTXT TEXT added.
```

To make the TXTLIB available for use, issue:

```
global txtlib yourmemb
```

Using VMFNLS to Update National Language Source Files

You use VMFNLS to apply updates to national language files and to compile the updated versions. Refer to "VMFNLS EXEC" on page 15-67 and "How VMFNLS Works" on page 15-67 for details about this procedure.

Using VMFLOAD to Punch a New Nucleus

After you have reassembled all the modules that require updating, you may build a new CP nucleus that contains the updated text decks. In our example, you also want to include your new module, DMKCMD, in the CP nucleus.

To build a new CP nucleus, you should select the build process that bests suits your installation. When using:

- SPGEN directly, refer to "SPGEN EXEC" on page 15-24.
- VMFLOAD directly, continue reading.

Since all the nucleus build processes use the VMFLOAD command, the following example details the VMFLOAD procedure. Refer to "VMFLOAD MODULE" on page 15-53 for a description of the VMFLOAD EXEC command format and more information about this EXEC.

VMFLOAD requires:

- A loadlist file, which must have a filetype of EXEC (in the following example we use the file called YOURLOAD EXEC). It contains the filenames of the object modules in the order they are to reside in the nucleus.
- A control file that lists update files or auxiliary control files (in the following example we use a control file called YOUROWN CNTRL). VMFLOAD uses these update and auxiliary control files to determine the latest level text decks, so it can punch them.

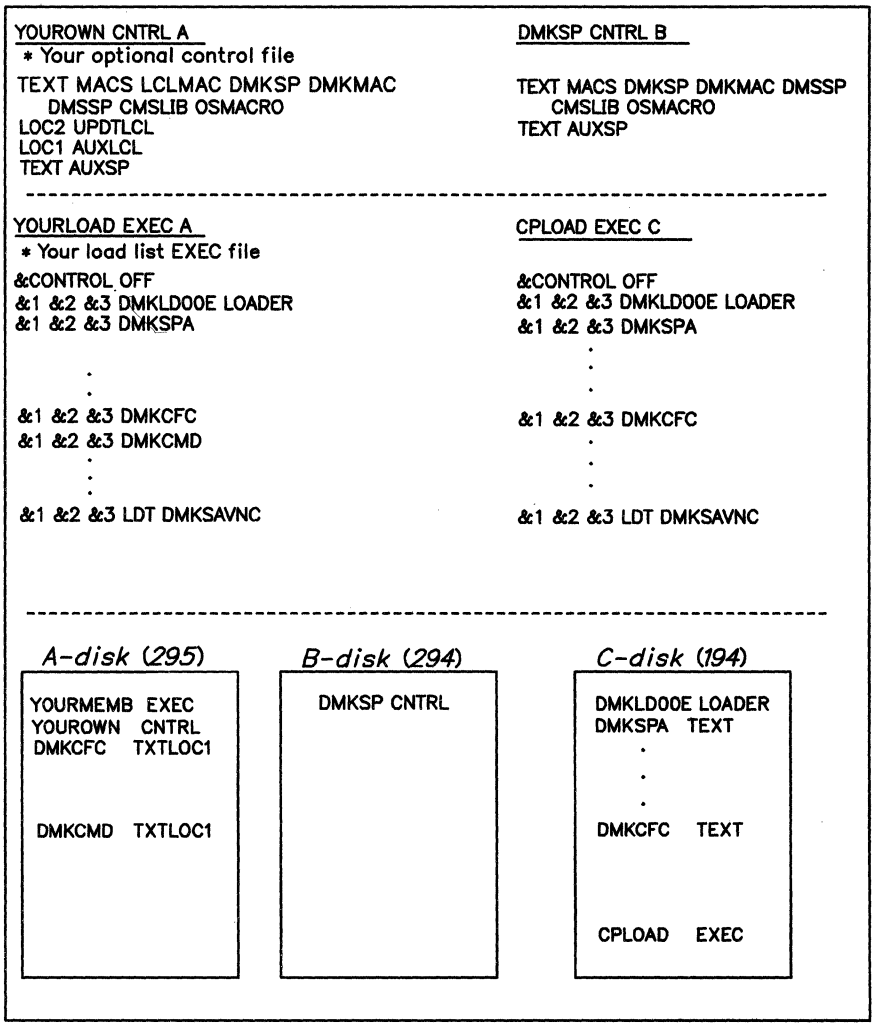


Figure 12-6. Files for VMFLOAD

Figure 12-6 shows the files used for creating a new CP nucleus. This nucleus incorporates the fixes described on the previous pages. The search order to build is:

295 A R/W
294 B/A R/O
194 C/A R/O
190 S R/O

Since the VMFLOAD command uses your virtual card reader and virtual punch, you must be sure there are no files in either of these devices before you begin. If you want to save any of the files in your punch or reader, load them to your disk. Otherwise issue the following commands to erase any files in your punch or reader:

close punch
purge punch all

close reader
purge reader all

You must also spool your virtual punch to your own card reader. Issue:

spool punch *

Then issue the command:

vmfload yourload [yourown]

The VMFLOAD command procedure uses YOURLOAD EXEC to determine which files to punch. In this example, YOURLOAD EXEC is identical to the distributed CPLOAD EXEC file, except for the added entry for your DMKCMD file.

- VMFLOAD takes the files from the loadlist (YOURLOAD EXEC) and punches them. They are punched in the order they appear in the loadlist. Thus, in this example, VMFLOAD punches DMKLD00E LOADER first.

  If you specify a filetype in the loadlist - as is the case with DMKLD00E LOADER - then VMFLOAD punches that specific file). Normally you do not specify a filetype, in which case VMFLOAD uses the update level identifier in the control file (in our example, the identifiers are LOC2, LOC1, and TEXT) to determine the filetype.

- DMKSPA, the next entry in the loadlist, does not have a filetype, thus VMFLOAD looks at the control file. Since control files are structured so that the most recent update is at the top of the file, VMFLOAD begins reading at the top of the file. In the example, the update level identifier for the first update record is LOC2.

- VMFLOAD searches for the file DMSPSA TXTLOC2. This file does not exist on any of the accessed disks.

- VMFLOAD looks at the next identifier, LOC1, and searches for DMKSPA TXTLOC1. This file does not exist on any of the accessed disks either.

- VMFLOAD looks at the next identifier, TEXT, and searches for DMKSPA TEXT. This file exists on the 194 disk, so VMFLOAD punches it.

- Then VMFLOAD looks at the next entry in the loadlist and repeats the entire process.

  You can see that when VMFLOAD reaches the entry for DMKCFC in the loadlist, it locates the file DMKCFC TXTLOC1 (on the 295 disk). The DMKCFC module contains your updates (Note that VMFLOAD takes the updates from the 295 disk and not from the DMKCFC TEXT on the 194 disk. This is because we have 295 accessed before 194 in the search order).

The loading process continues in this way until VMFLOAD reaches the end of the loadlist EXEC file. When all the modules have been punched, you receive the messages:

```
SYSTEM LOAD DECK COMPLETE
```

```
PUN FILE nnnn TO MAINT COPY 001 NOHOLD
```

These messages indicate that a copy of the new CP nucleus is in your card reader. This CP nucleus contains all the text decks on the 194 disk, except that the files:

DMKCFC TXTLOC1
DMKACO TXTLOC2
DMKRIO TXTLOC1
DMKCMD TXTLOC1

have been punched instead of their TEXT counterparts.

Once the new nucleus has been punched into your card reader, you can load it and test it. Refer to Appendix K, "Building CP and CMS Nuclei Using SPGEN" on page K-1 details.

To build:

- GCS nucleus, run the SPGEN EXEC with NOSETUP option.

  If you have applied service to any GCS MACLIBs, then during a GCS build you need to copy your GCS GROUP file to an ASSEMBLE filetype. Then, run SPGEN to reassemble the ASSEMBLE file before doing the GCS nucleus build.

- TSAF module, run the GENTSAF EXEC.

- CMS modules, run the CMSGEND EXEC.

- IPCS modules, run the GENMOD EXEC.

With the nucleus and module builds complete, you need to build any modules or discontiguous saved segments (DCSS) that received service. Refer to Chapter 6, "Installing Discontiguous Saved Segments" on page 6-1 for details.

Apply Preventive Service

For IBM products, you receive Program Update Tapes (PUTs) containing replacement files. The following describes the PUT.

PUT Format

First Physical Tape File

The first physical tape file on each volume of the Program Update Tape (PUT) contains the PUT HEADER. This file consists of four CMS files which include:

PUT level	(i.e., PUT 8503)
PUT DOCUMENT	describes the format and installation of the PUT
VMSERV EXEC	is the user interface for applying service contained on the PUT.
VMFSTK MODULE	places a specific line into the console input buffer.

Second Physical Tape File

The second physical tape file of the PUT is the PUT MEMO file. This file consists of two CMS files for each product whose service follows on the PUT:

- pnum crllnn
- pnum MEMO

where:

pnum is the product number of the product (e.g., 5664167)

crllnn is the level identifier

- *c* is the corequisite flag
- *r* is the release level
- *ll* is the service level
- *nn* is the number of service files for the product.

pnum MEMO is the Memo-to-Users file describing user installation information

Third Physical Tape File

The third physical tape file of the PUT contains the first product's service files.

The first product's service file contains two CMS files:

- pnum crllnn
- pnum EXEC

where:

pnum EXEC is the service EXEC for the product.

The product's structure and service requirements determine the number and content of the succeeding service related files.

Service for each product on the PUT consists of two or more physical files.

Sample Layout of a PUT

Figure 12-7 shows a sample file layout of a VM PUT volume containing 2 products (represented by pnum(1) and pnum(2)).

File	Content
1	PUT HEADER File: PUT yysn (Year/Sequence Number) PUT DOCUMENT VMSERV EXEC VMFSTK MODULE
2	PUT MEMO File: pnum(1) crllnn pnum(1) MEMO pnum(2) crllnn pnum(2) MEMO
3	Service EXEC File: pnum(1) crllnn pnum(1) EXEC
4 to M	Service for the first product pnum(1) on this volume (one or more files) M is the last service file for the first product pnum(1).
M+1	Service EXEC File: pnum(2) crllnn pnum(2) EXEC
M+2 to XX	Service for the last product PNUM(2) on this volume (one or more files) XX is the last service file for the last product pnum(2).

Figure 12-7. Sample VM PUT Volume Layout

Note that all files on the tape are in VMFPLC2 format. Refer to page 15-72 for more information about VMFPLC2.

To apply preventive service, you need to:

1. Read the Memo(s)-to-Users and all other service documentation sent to you with your service tape.

   Any PUT Memo(s)-to-Users information replaces or supplements information contained in this section. Be sure to review it before applying service.

2. Call your IBM support center for the latest service information for the type of service you are doing.

   You may want to request a PSP bucket, i.e., a tape containing the latest service since the PUT was built. When this tape arrives, mount it, load and print the first tape file (the service documentation related to this tape). Read the documentation to decided which fixes to apply to your environment.

   Note: You should apply the fixes from the PSP bucket after you apply the service from the PUT to ensure you have the latest level of service.

3. Make sure you have a good backup copy of your current system.

4. Make sure you have the correct minidisk search order sequence if you are not using the defaults.

5. Load the service from the tape.

If you have not already done so, log onto the MAINT userid. Mount the service tape (set to read only mode) and issue the following commands:

vary on rdev

rdev is the real address of the service tape device. If you are running second level, you need to specify the virtual address of the service tape device.

attach rdev * 181

access 191 c

vmfplc2 load * * c

vmserv

The VMSERV EXEC scans the service tape and allows you to print the PUT document and Memo(s)-to-Users.

```
.
.
.
SHOULD THESE 'MEMO(S)-TO-USERS' BE PRINTED? - (-YES- | NO)
```

yes

VMSERV then prints the PUT DOCUMENT and Memo(s)-to-Users, and reminds you to read them before installing service. After you review the documentation and contact IBM about the latest service activity, use the VMSERV EXEC to load service for base VM/SP as described in the PUT DOCUMENT and Memo(s)-to-Users.

Once the service is loaded you need to do the `build` processes.

- Access minidisks with the correct search order (as shown in "Disk Addresses for Building" on page 11-7).

- Run the appropriate exec procedure to do the build.

  - **Run VMSERV** which calls the service exec

    To build each component of VM/SP, reinvoke VMSERV using disk addresses from the VMPUT PRODUCTS file (if it exists and is complete) or respond to prompts. The prompts allow you to select addresses or to accept the service exec minidisk address defaults.

  - **Run the Service Exec** which calls SPGEN with the NOSETUP option for all nucleus builds.

    To build each component of VM/SP, reinvoke the service exec supplying disk addresses as input or respond to prompts. The prompts allow you to select addresses or to accept the service exec minidisk address defaults.

  - **Run SPGEN** which calls VMFLOAD.

    To build nuclei (CP, CMS, and GCS) for VM/SP, reinvoke SPGEN using the disk addresses contained in the SPGEN PROFILE file.

  - To build:

    - TSAF module, run the GENTSAF EXEC.

    - CMS modules, run the CMSGEND EXEC.

    - IPCS modules, run the GENMOD EXEC.

  If you have applied service to any GCS MACLIBs, then during a GCS build you need to copy your GCS GROUP file to an ASSEMBLE filetype. Then invoke SPGEN to reassemble the ASSEMBLE file before doing the GCS nucleus build.

- Move the nuclei to disk(s) and IPL the test system. Refer to Appendix K, "Building CP and CMS Nuclei Using SPGEN" on page K-1 for more information about this step.

- Do a full system backup.

Chapter 13. Servicing Object Maintained VM/SP Products

This chapter contains information about how to service products by full replacements.

Apply Corrective Service to CMS Object, GCS, and TSAF

After you order service from the IBM Support Center, you receive a service tape and a service document. The first file on the tape contains a copy of the service document. The service document explains the contents of the service tape and the instructions for loading the service from tape to disk. Files 2 to n contains the service files. The procedure for loading the service to disk is a follows:

1. Create a good backup copy of your system.

2. Mount and attach the service tape.

3. Access the disk to receive service as A.

   We recommend that you use some other disk than the default service minidisk(s), those being 293, 596, and 494 for CMS, GCS and TSAF respectively. Reserve these minidisks for service from the PUT. Use some other disk for corrective service.

   Receive and load the PUT contained the service to the component's respective service minidisks. Verify that the PUT contains all the fixes and then erase the corrective service from the other disk. This helps you keep track of the service you apply in a more orderly fashion.

4. Use VMFPLC2 to load the contents of the corrective service tape.

   The service document explains the contents of the tape and how many files there are. If you do not want to load the entire tape, use the SCAN option of the VMFPLC2 command to find the file(s) you want to load. Refer to "VMFPLC2 MODULE" on page 15-72 for more information about VMFPLC2.

5. Run the product rename exec supplied on the tape.. The exec is in the file named 5664167D EXEC[2] You must specify the disk containing the

---

1 The VM/SP service exec used by VMSERV to apply preventive service and the rename exec you use when applying corrective service, are two *different* execs but they have the same name -- 5664167D EXEC

service as part of the command when you run the exec. (Refer to the corrective service tape cover letter for more details about this rename exec).

The rename exec renames TXT files to TEXT and erases all other files not renamed.

Notes:

a. *If you have updates to both CMS source and object, refer to "Apply Corrective Service to Source Code" on page 12-8 for the procedure to apply the source updates.*

b. *If you need to apply updates to national language files, refer to "VMFNLS EXEC" on page 15-67.*

6. Build the new nuclei/module.
For **CMS**:

   - build any MACLIBs that received service. Use the VMFMAC EXEC procedure described in "Using VMFMAC to Update Macro Libraries" on page 12-11.

   - apply any source updates and reassemble the changed files. Refer to "Apply Corrective Service to Source Code" on page 12-8 for the procedures to use.

   - access the needed minidisks. The recommended search order is:

     user modifications minidisk
     corrective service minidisk
     CMS service minidisk
     CMS base minidisk
     CMS system minidisk

   - build the CMS nucleus and read in the load map using SPGEN. Refer to "Building the New CMS Nucleus" on page K-3.

   - build updated CMS modules. Refer to "Updating CMS Modules" on page 14-9.

   - build any discontiguous saved segments (DCSS) that received service. Refer to "Updating Shared Segments" on page 14-12.

   For **GCS**:

   - access the needed minidisks. The recommended search order is:

     user modifications minidisk
     corrective service minidisk
     GCS service minidisk
     GCS base minidisk

CMS service minidisk
CMS base minidisk
CMS system minidisk

- reassemble the GROUP file if you applied any service to any GCS MACLIBs. (Use "SPGEN EXEC" on page 15-24 or "VMFASM EXEC" on page 15-38).

- build the GCS nucleus and read in the nucleus load map using "SPGEN EXEC" on page 15-24 or "VMFLOAD MODULE" on page 15-53.

For **TSAF**:

- access the needed minidisks. The recommended search order is:

  user modifications minidisk
  corrective service minidisk
  TSAF service minidisk
  TSAF base minidisk
  CMS system minidisk

- build the TSAF module using the GENTSAF EXEC procedure. Refer to "GENTSAF EXEC" on page 15-11.

Apply Preventive Service to Object Code

To apply preventive service to object code, you follow the same procedure as for source code. Refer to "Apply Preventive Service" on page 12-27 for the procedure.

Chapter 14. Updating Service Programs, Modules, and Shared Segments

This chapter discusses general considerations and recommended procedures for updating:

- Service programs
- CMS Modules
- Printer Modules
- Programmable Operator Modules
- Interactive Problem Control System (IPCS)
- VSE/VSAM and CMSDOS shared segments.

When service is shipped via PUT, all modules are regenerated and replaced on a one-to-one basis. Whenever CMS source code is updated, the CMS nucleus and/or some CMS modules/segments must be regenerated. Refer to Appendix M, "Regenerating CP/CMS/IPCS Modules/Segments" on page M-1 for more information about which modules/segments must be regenerated.

Updating Service Programs

Service programs are CP modules that are not a part of the CP nucleus. They may run either stand-alone from a card reader (the real system card reader or your virtual card reader), or in some cases as a CMS command. The service programs are:

- DASD Dump/Restore (module DMKDDR)
- Directory program (module DMKDIR)
- Format/Allocate program (module DMKFMT)
- Device Support Facility
- NCPDUMP, the 3704/3705 dump program (module DMKRND)
- Loader (DMKLD00E).

In most cases, if you apply a PTF to a service program, you may use the UTILITY EXEC to create a new IPLable copy of the service program for loading via IPL or the CMSGEND EXEC to create a new CMS command module, or both. Figure 14-1 on page 14-2 indicates the procedures you can use to update each program.

When using the UTILITY EXEC, the text decks being used to create new IPLable copies must reside on the CMS system disk.

Service Program	CP Source (use VMFASM to update)	CMS Command (use CMSGEND to update)	CMS Disk File (use UTILITY IPLDECK to update
DASD Dump/ Restore	DMKDDR DMKDNC DMKDNT DMKDDC DMKDDT	DDR	IPL DDR
Directory	DMKDIR	DIRECT	IPL DDR
Format/ Allocate	DMKFMT	—	IPL FMT
NCPDUMP	DMKRND	NCPDUMP	—
Device Support Facility	N/A	N/A	N/A
Loader	N/A	N/A	N/A

Figure 14-1. Updating Service Programs

For example, the Directory program exists as the CP module DMKDIR. To apply PTFs to the source file, you would use the VMFASM EXEC procedure, as follows:

vmfasm dmkdir dmksp

dmksp is the filename of the control file (the filetype is CNTRL).

The Directory program can be used as:

- A stand-alone program that you IPL from the real system card reader
- A stand-alone program that you can IPL from your virtual card reader
- A CMS command, DIRECT.

To create a new stand-alone copy via IPL, use the UTILITY EXEC as follows:

utility ipldeck dir

The UTILITY EXEC prompts you to enter the target disk address where the service programs will be written.

Type 190 (or vdev)

The UTILITY EXEC is finished.

To generate the CMS DIRECT command, use the CMSGEND EXEC procedure as follows:

cmsgend direct

Updating the Loader Program

The loader (DMKLD00E) is a service program that loads the CP nucleus, CMS nucleus, or Stand-Alone Dump (SAD) program, and produces a load map. The loader loads the object modules (text files) supplied with it, resolves CCW addresses, and resolves address constants.

If an overlay error occurs while the loader is executing, define a larger virtual machine and reload the system.

The loader is distributed with the following default I/O addresses:

```
Console=009[1]
Printer=00E[2]
```

If there is no printer at address 00E, the load map is printed at the first printer that causes an interrupt (not-ready to ready sequence). A reply of 'ignore' can be made to the printer intervention required message from the loader. This allows processing to continue without the use of the printer.

---

1 The loader does not support display mode consoles. If an IPL is attempted, wait state code X'FFF' occurs if the printer address is not 00E. To circumvent this occurrence, reconfigure the console to printer-keyboard mode or use the override procedure to correct the printer address. Refer to the *VM/SP Planning Guide and Reference* for information about the override procedure.

2 If the printer address is not 00E, IPLing the loader results in a wait state code of X'FFF'.

Overriding the Default I/O Addresses

You can override the default I/O addresses by placing a control card between the last card of the loader and the first card of the text decks. The format of the control card is:

Column	Contents
1	12-2-9 multipunch (XX'02')
2-4	DEV
5	Blank
6-13	PRNT = vdev (vdev is the printer address)
14	Blank or comma
15-22	TYPW = vdev (vdev is the console address)
23-72	Blank

The other loader control statements are the same as the loader control statements described with the CMS LOAD command in the *VM/SP CMS Command Reference*.

The loader is self-relocating; that is, it is initially loaded at address 2000 (hexadecimal); it then relocates itself at the top of storage. (For example, if the size of the loader is 10K, and the real storage size of the processor is 512K, the loader occupies the area of storage between 502K and 512K.) As the loader needs free storage to do its operations, it extends downward through storage.

The object modules being loaded must not overlay either the loader or any address between 0 and 100 (hexadecimal). The object modules are loaded into storage in a positive direction (that is, upward through storage). Before the loader actually loads an object module, it checks that the module does not overlay the loader's free storage. If an object module would overlay the loader, the loader terminates. You must close the printer to get the load map printed. The last line of the load map indicates the overlay area, if there was one.

If the loader terminates the operation, a wait condition is indicated in the instruction counter. If the instruction counter contains X'999999', indicating an SVC wait state, the interruption code (the third and fourth bytes of the supervisor old PSW) indicate the error condition. For a detailed explanation of the error conditions and interruption codes, see *VM/SP System Messages and Codes*.

The Load Map

The load map (the output of the loader) indicates:

- The size of each object module and the address where it is loaded. For example:

```
DMKMCH AT 00E68   MODULE SIZE IS 000C00
```

- The end of the resident nucleus with the message:

```
***                                        ***
      END OF VM/SP  RESIDENT NUCLEUS
***                                        ***
```

  The CP modules that precede this message in the load map are not pageable; the CP modules that follow this message are pageable.

- When a Set Page Boundary (SPB) card has been inserted. If an object module cannot fit on the same page as the object module(s) loaded before it, the loader inserts an SPB card to force the modules to be loaded at a page boundary. This procedure ensures that object modules do not cross page boundaries.

- Two external names may be listed as undefined on the load map. If the virtual=real option is not specified, the external name DMKSLC is listed as undefined. If a 3704/3705 control program entry is not defined in the system name table (via the NAMENCP macro), the DMKSNTRN external name is undefined.

Generating a New Loader

The loader service program, in its executable form, has a filetype of LOADER. Whenever you assemble a new copy of DMKLD00E, you must convert the resulting text file to a loader file. If there is a virtual punch at address 00D and a virtual reader at address 00C, the procedure for generating a new loader is:

Step 1. Assemble the New Loader

Update and assemble DMKLD00E. The output from this assembly is DMKLD00E TEXT.

Step 2. Punch a Copy of the Old Loader

To spool the punch continuous and punch a copy of the old loader, issue:

```
spool 00d * cont
```

punch dmkld00e loader (noh

Step 3. Punch a Copy of the New Loader Text File

Punch a copy of the newly assembled loader, then close the punch. When the punch is closed the two files (DMKLD00E LOADER and DMKLD00E TEXT) are sent to your reader. The commands to punch the new loader text file and close the punch file are:

punch dmkld00e text (noh

spool 00d * nocont close

Step 4. Load the New Loader

IPL your virtual reader to read the old version of the loader (DMKLD00E LOADER) into your virtual machine. The old loader reads the new loader text file into your virtual machine and creates the new loader file.

ipl 00c clear

```
DMKDSP450W CP entered; disabled wait PSW X'404040'
```

IPL is complete. The instruction address in the disabled wait PSW is X'404040'.

Step 5. Punch a Copy of the New Loader (Executable Form)

Close the punch to punch a copy of the new loader, which was created in Step 4. Also, close the reader and printer by issuing:

close 00d

close 00c

close 00e

Step 6. Name the New Loader DMKLD00E LOADER

IPL CMS and access a read/write disk as your A-disk.

Note: Save a copy of your original DMKLD00E LOADER file before you replace it below with the updated loader. Then, read the file you punched in Step 5 by entering:

ipl cms

readcard dmkld00e loader

By naming this updated file DMKLD00E LOADER, it replaces the original DMKLD00E LOADER file.

Updating CMS Modules

To update CMS modules, you need to copy all files with the filetypes or TEXT and MODULE onto a new disk from the 190 system disk and regenerate any modules that use auxiliary directories (such as the ASSEMBLE command). Auxiliary directories are described in the *VM/SP CMS for System Programming*. You can use the CMSGEND EXEC procedure to regenerate the assembler. Some IBM Licensed Programs may also use auxiliary directories.

Before you go on to build the new nucleus, you need to regenerate any CMS MODULE files that have been updated.Appendix M, "Regenerating CP/CMS/IPCS Modules/Segments" on page M-1 contains information about:

- creating and building CMS disk resident modules
- CP/CMS/IPCS Module/Segment Regeneration Requirements

To decide how to put a given CMS textfile into production, look for its name in the CMSGEND EXEC and the CMSLOAD EXEC (the CMS nucleus loadlist). If the textfile is listed in the CMSGEND EXEC, then it must be incorporated into a CMS MODULE via CMSGEND. If the name is listed in the CMSLOAD EXEC, then the textfile is part of the nucleus; you install it by rebuilding the nucleus.

Updating Printer Modules

Modify the following as required to meet your installation's needs. See module prologues for specific information.

- Printer Universal Character Set
  DMKUCS (1403)
  DMKUCB (3211)
  DMKUCC (3203)
  DMKPIB (3262)

- Font Offset Buffer
  DMKPIA (3289)

- Forms Control Macros
  DMKFCB (forms control)

To alter any of the printer modules previously listed, enter the following:

access 294 a

access 194 b

access 394 c

copy dmk*xxx* assemble c = = a

xxx is the appropriate module suffix as listed above.

If the CP source is not on the 394 minidisk, mount the tape containing CP source on a tape drive at virtual address 181.

Note: For both 1600 and 6250 bpi tapes, CP source is the first file on the tape.

Load DMK*xxx* by issuing:

access 294 a

vmfplc2 load dmk*xxx* assemble (eot

To modify the file to include your changes enter the following:

xedit dmk*xxx* assemble

Make the appropriate changes and store the file. Then, assemble DMK*xxx* by entering:

vmfasm dmk*xxx* dmksp The DMK*xxx* file assembles at this time.

If any assembly errors occur, they need to be corrected before the changes you made can be used.

Updating Programmable Operator Modules

You need to access your 294 and 194 minidisks to gain access to the latest version of DMKSP MACLIB in order to assemble the following programmable operator modules:

- DMSPOA
- DMSPOC
- DMSPOD
- DMSPOE
- DMSPOL
- DMSPON
- DMSPOP
- DMSPOQ
- DMSPOR
- DMSPOS

Refer to *VM System Facilities for Programming* for more information about the programmable operator facility.

Updating Interactive Problem Control System (IPCS)

Any program modifications or service updates to IPCS are distributed via a service tape.

Since the service tape includes pre-assembled text files, it is necessary to assemble only those modules that your installation has changed. Once you load the updated files to disk, you can use VMFASM to assemble any source files that you have changed. For example:

vmfasm dmmpro dmmsp

where "dmmpro" is the name of the file to be reassembled, and "dmmsp" is the name of the control file to be used.

See Figure M-4 on page M-11 to find out which source files should be regenerated.

To generate the new IPCS command modules, use the LOAD and GENMOD commands. Refer to the *VM/SP CMS Command Reference* for details on

the LOAD and GENMOD commands. Once the new IPCS module has been generated, copy it to the MAINT 193 minidisk.

Updating Shared Segments

In this section we discuss updating the VSE/VSAM, CMSBAM, and CMSDOS shared segments.

Updating VSE/VSAM

You are responsible for applying service for VSE/VSAM from the VM System Program Update Tape (PUT). This tailored service tape contains VSE/VSAM PTFs in the form of CMS TEXT files and the service EXEC used for applying the service to the CMSVSAM and CMSAMS segments. The master service EXEC (VMSERV) that resides on the service tape invokes the VSAM service EXEC to apply the PTFs. The EXEC is self-documenting and will prompt you for all information required to restore the updates from the tape and/or rebuild the segments. Refer to page 15-91 for more details about the VMSERV EXEC.

If you wish to rebuild the segments after service is loaded, you must have available either a minidisk containing the VSE/VSAM text files that were previously restored from the VSE/VSAM installation tape, or the VSE/VSAM installation tape attached as virtual 181, and a minidisk large enough to hold all the text files. There must also be enough virtual storage defined to allow the various phases to be fetched into storage from the DOSLIBs. If you have followed the example on preparing the System Name Table file (DMKSNT), 15360K is needed.

VSE/VSAM Update Considerations

Applying VSE/VSAM PTFs to either CMSVSAM or the CMSAMS discontiguous saved segments may result in the generated segment exceeding the space defined for it in the system name table (see the NAMESYS macro of the DMKSNT file in the *VM/SP Planning Guide and Reference*). You may want to anticipate this problem by defining in the system name table an additional shared and nonshared segment for each of the discontiguous saved segments (CMSVSAM and CMSAMS). This is one way of providing for additional growth.

Alternatively, on completion of the update procedure, you can check whether the updated segments have exceeded their definitions and correct that situation as follows:

1. Determine the new size of the changed VSAM and/or Access Method Services shared and nonshared segments by subtracting the phase

LOCORE address from the HICORE address indicated on the linkage editor map. The phase names are:

- DMSVVS - VSAM shared
- DMSVVN - VSAM nonshared
- DMSVAS - Access Method Services shared
- DMSVAN - Access Method Services nonshared
- DMSVAX - Access Method Services nonshared

2. Compare the new sizes of these segments with the sizes of the corresponding shared or non-shared segments as defined in your DMKSNT NAMESYS macro.

3. If the new size exceeds your defined size, recode the NAMESYS macro to include an additional segment. Refer to the phase names listed in Step1 to determine whether the segment is shared or nonshared. To add one segment:

   - Increase the SYSPGCT operand by 16
   - Increase the SYSPGNUM operand by 16
   - Increase the SYSHRSG operand by 1, if the segment is shared
   - Increase the SYSSIZE operand by 64K
   - Change the SYSSTRT operand of this or other segments, if the increase in this segment causes any segment to overlap.

4. Reassemble the DMKSNT file, build a new CP nucleus, and then re-execute the VSAMGEN procedure.

If a PTF contains a new VSAM or Access Method Services module, it is not included in CMSVSAM or CMSAMS during VSAMGEN unless you have the current level of installation files.

Updating CMSDOS and CMSBAM

If you need to update the CMSDOS or CMSBAM discontiguous saved segments, then you must rebuild them in their entirety. Refer to "Installing CMSDOS" on page 6-6 and to "Installing CMSBAM" on page 6-9 for the procedures to rebuild these segments.

Part 4. VM/SP Installation/Service EXECs and Commands

Chapter 15. EXEC and Command Procedures

ASMGEND EXEC

The ASMGEND EXEC procedure builds the system assembler and creates the associated auxiliary directory. ASMGEND loads the text decks for the assembler in the correct overlay structure and produces a load map.

Command Format

ASMGEND	

Usage Notes

1. The assembler text decks normally reside on the system S-disk in file mode S1. This disk must be accessed in some additional file mode prior to issuing this command in order to locate these files. For example:

   access 190 a
   access 193 b

2. Use the ASMGEND EXEC if you have modified the assembler (IFOX*nn*) source. If you have not modified this source, and wish to create the assemble module, possibly after modifying DMSASM, DMSASN, DMSASD, or creating a new CMS system disk, use the CMSGEND EXEC.

Messages

ENTER TARGET DISK MODE FOR ASSEMBLE MODULES
DEFAULTS TO S-DISK IF NONE ENTERED

Enter the mode letter of the disk containing the assembler modules. The ASSEMBLE command accesses this disk during processing. If you enter a mode letter, ASMGEND uses that mode letter as the "targetmode" operand of the GENDIRT command when it creates the auxiliary directory. If you do not specify a mode letter, S is used.

ASSEMBLE XF GEND COMPLETE

The system assembler and its associated auxiliary directory have been generated successfully.

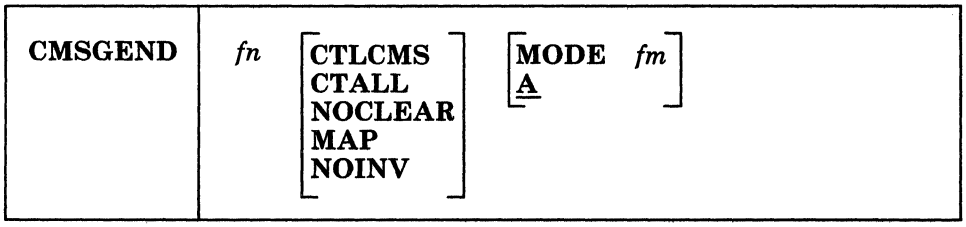
ASSEMBLE XF GEND FAILED

The system assembler text files were not loaded successfully.

CMSGEND EXEC

The CMSGEND EXEC procedure generates a new CMS module or LOADLIB from a text file and places the new CMS module or LOADLIB on the specified disk.

Command Format



fn
is the filename of the CMS module or LOADLIB that is to be generated by the CMSGEND EXEC. Only one filename may be specified in the CMSGEND command line.

The filenames that may be specified in the CMSGEND command are any disk-resident CMS commands and service programs.

CTLCMS
displays each CMS command as it is executed in the CMSGEND EXEC procedure. This is equivalent to the EXEC statement &CONTROL CMS.

CTALL
displays every executable statement as it is executed in the CMSGEND EXEC procedure. This is equivalent to the EXEC statement &CONTROL ALL.

NOCLEAR
specifies that the CLEAR option is not to be issued when CMSGEND invokes the LOAD command.

MAP
specifies that the NOMAP option is not to be issued when CMSGEND invokes the GENMOD command.

NOINV
issues the NOINV option when CMSGEND invokes the LOAD command; this suppresses the displaying of invalid cards at the terminal. If the text deck was created with the VMFASM EXEC, it may contain update listing information; these records are displayed during the loading process unless you specify NOINV.

MODE *fm*
indicates the access mode of the disk to receive the new module. Filemode A is the default.

How CMSGEND Works

CMSGEND keeps a list of the CMS disk-resident modules and LOADLIBs, the filenames of the text files used to create them, and any special attributes required to generate them. For example, the PRINT command must be generated with the ORIGIN TRANS and the SYSTEM options. It is composed of the DMSPRT text file. To generate a new PRINT module, you issue:

cmsgend print

```
*** CURRENT STATUS:
FILE ' PRINT MODULE A2 ' DOES NOT EXIST
FILE ' PRINT MODULE A1 ' DOES NOT EXIST

*** LOADING:
INVALID CARD - *  DMSSP  MACLIB  A1 5.0CMS mm/dd/yy hh:mm:ss
        .
        .
        .
DMSPRT     S0 00E000
PRINT         00E000

***RESULTS:
' PRINT MODULE A2 ' CREATED FROM TEXT DECK ( S ) DMSPRT

WITH OPTIONS TRAMX SYSTEM NOMAP

Ready;
```

Usage Notes

1. The assembler text decks normally reside on the system S-disk in file mode S1. This disk must be accessed in some additional file mode prior to issuing this command in order to locate these files. For example:

   access 190 a (CMS system disk)
   access 293 b (CMS service disk)
   access 193 c (CMS base disk)

2. Use the CMSGEND EXEC if you have not modified the assembler (IFOX*nn*) source, and wish to create the assemble module, possibly after modifying DMSASM DMSASN, DMSASD, or creating a new CMS system disk.

3. You can also use the CMSGEND EXEC to regenerate the ASSEMBLE command when you move the CMS system disk. When you specify ASSEMBLE, CMSGEND prompts you to enter a disk mode letter so it can refresh the assembler's auxiliary directory. (Use the ASMGEND EXEC procedure if you are updating the assembler.)

4. When using CMSGEND EXEC to regenerate the PROP command, you are only generating the PROPLIB LOADLIB. (The CMSGEND options NOCLEAR, MAP, and NOINV have no effect when generating the PROP command.)

Messages

```
*** CURRENT STATUS:

   ['fn MODULE A2'    EXISTS        ]
   ['fn MODULE A2'    DOES NOT EXIST]
   ['fn MODOLD A1'    EXISTS        ]
   ['fn MODOLD A1'    DOES NOT EXIST]
```

This response indicates whether or not a generated module already exists.

```
*** LINK EDITING: fn TEXT
```

CMSGEND is link editing *fn* TEXT to create a LOADLIB. The existing LOADLIB is erased and not renamed when generating a new one.

```
*** LOADING:
```

CMSGEND is loading the text decks.

```
*** (UNDEF. NAMES NORMAL FOR EDMAIN)
*** NOW WE HAVE A SECOND PASS FOR EDMAIN MODULE.
```

These responses indicate that the EDIT command requires two passes to resolve undefined names.

```
*** NOW WE HAVE A SECOND PASS FOR DMSTPI MODULE.
```

This response indicates that the TAPE command requires two passes to generate the necessary modules. The first pass generates the DMSLMX module as the TAPE module; the second pass generates the DMSTPE, DMSTPG, DMSTPH, DMSTPI, and DMSTPJ text files as the DMSTPI module.

```
*** RESULTS:
   ['fn MODOLD A1' WAS ERASED              ]
   ['fn MODULE A2' RENAMED TO 'fn MODOLD A1']

   'fn MODULE A2' CREATED FROM TEXT DECK(S) ...
 WITH OPTIONS ...
```

These responses indicate which existing modules were erased and renamed, which text files were used to create the new module, and the attributes used to create the module.

ENTER GENDIRT TARGET DISK MODE LETTER
(NULL LINE DEFAULTS TO 'S' DISK)

This prompt is issued when you specify the ASSEMBLE module. Enter the mode letter of the disk that contains the system assembler. This letter is used as the target disk mode address for the GENDIRT command.

***** ERROR MESSAGE ISSUED IS NORMAL FOR LINK EDITING**

If the TEXT deck was created with VMFASM EXEC, it may contain update listing information. These records will cause the linkage editor to generate an error message. The error is normal.

ERROR OCCURRED. CMSGEND STOPS.

An error has occurred and CMSGEND processing has stopped.

INVALID ARGUMENT fn

You specified an invalid filename on the command line.

TYPE 'CMSGEND fn <options>'

You did not specify a filename on the command line. Reissue the command with a filename and the operand(s) you want to use.

DIRECT MODULE

The DIRECT command processes the control statements in a directory file.

Command Format

DIRECT	[*fn* / **USER** [*ft* / **DIRECT** [*fm* / *****]]] [**(EDIT)**]

fn
is the filename of the directory file. The default is USER.

ft
is the filetype of the directory file. The default is DIRECT.

fm
is the filemode of the directory file. The default is *.

(EDIT)
specifies that the directory file is to be examined, but not changed or swapped for the currently active directory.

How DIRECT Works

The DIRECT command loads the VM/SP directory program. The first statement in a directory file must be a DIRECTORY statement; otherwise, the program stops. A syntax error in any statement generates an error message, and the directory is not updated. If no critical errors are encountered, the remaining statements are checked for syntax. If the directory program abnormally ends, the old directory is not altered. Normal completion of the program places the directory in use by VM/SP. For more information about the directory program and directory control statements, refer to the *VM/SP Planning Guide and Reference.*

Usage Notes

1. If you want to change or swap the currently active directory, you must have write access to the system-owned (system residence or IPL device) volume that contains the current directory up to and including the directory cylinders, or have write access to the volume that is to contain the new directory.

2. If you have the above qualification and only want to verify that the directory file follows the required format, you must use the EDIT option. Otherwise, if there are no control statement errors, the file is put into active use.

3. If you are building a VM/SP directory on a CP-owned volume using preallocated cylinders, do not build the new directory to overlay the

existing directory. You must allow space for two directories, or allocate a new area for the new directory.

4. When you run the directory program, if the directory you are updating is currently in use by the system, or if the target directory pack is present in the system-owned list, then the new directory is dynamically swapped for the current directory and placed on line by the system.

5. When the directory is updated, directory changes for a user currently logged on to the system do not take effect until the user logs off the system and then logs back on.

6. When a new directory is written for a new system residence volume, the new directory does not take effect until the new system residence volume is IPLed.

DISKMAP EXEC

The DISKMAP EXEC summarizes the MDISK statements in a CP directory. The output produced by this EXEC shows gaps and overlaps between minidisk assignments.

Command Format

DISKMAP	*fn*	[*ft* / **DIRECT**]

fn
is the filename of the directory to be mapped.

ft
is the filetype of the directory to be mapped. The default is DIRECT.

How DISKMAP Works

The output from the DISKMAP EXEC is a file sent to your A-disk. The filename of the output file is the same as the filename of the target directory; the filetype is DISKMAP. In this file, information about the directory MDISK statements is organized by CP volume. Gaps between minidisks and overlapping minidisks are flagged.

To see how DISKMAP works, enter:

diskmap vmusers

This will produce a directory map of the sample files that you loaded during installation of VM/SP.

DISKMAP does NOT replace the EDIT function of the DIRECT command. You should use both to check your directory after changes. (The DIRECT command is described on page 15-7 and also in the *VM/SP Planning Guide and Reference.*)

Usage Notes

1. DISKMAP creates both the map and a workfile on your A-disk. If your directory is very large and your A-disk is almost filled, you may need to find some extra disk space in order to run DISKMAP.

2. Because some DASD types come in several sizes, DISKMAP does not list gaps found after all minidisks. You will need to know the maximum cylinder/block value for your DASD type.

3. You may choose to include some overlaps in the directory. DISKMAP flags ALL overlaps; you must understand your layout to determine if a particular overlap is expected or in error. For instance, in the VM/SP

sample directory, the same full-pack minidisks (12x) are defined for backup purposes for MAINT and SYSDUMP1.

GENTSAF EXEC

The GENTSAF EXEC procedure builds the RUNTSAF module and creates a TSAF load map.

Command Format

GENTSAF	*loadlist* [*ctlfile*]

loadlist
is the name of the loadlist containing all the TSAF text files. The filetype of the loadlist must be EXEC. The loadlist shipped with the system is named ATSLOAD EXEC.

ctlfile
is the filename of an optional control file. The filetype must be CNTRL. The IBM-supplied control file (ATSSP CNTRL) is described in Appendix L, "Control File Identifiers" on page L-1. If you specify a control file, the PRELOAD command uses the file as input. Refer to "PRELOAD MODULE" on page 15-18 for more details about PRELOAD processing using a control file.

How GENTSAF Works

The GENTSAF EXEC procedure PRELOADs all the text files listed in the loadlist and then LOADs the resulting TEXT files. Then the EXEC does a GENMOD. It produces the following files:

- RUNTSAF MODULE A5
- TSAF MAP A5
- *loadlist* MAP A5

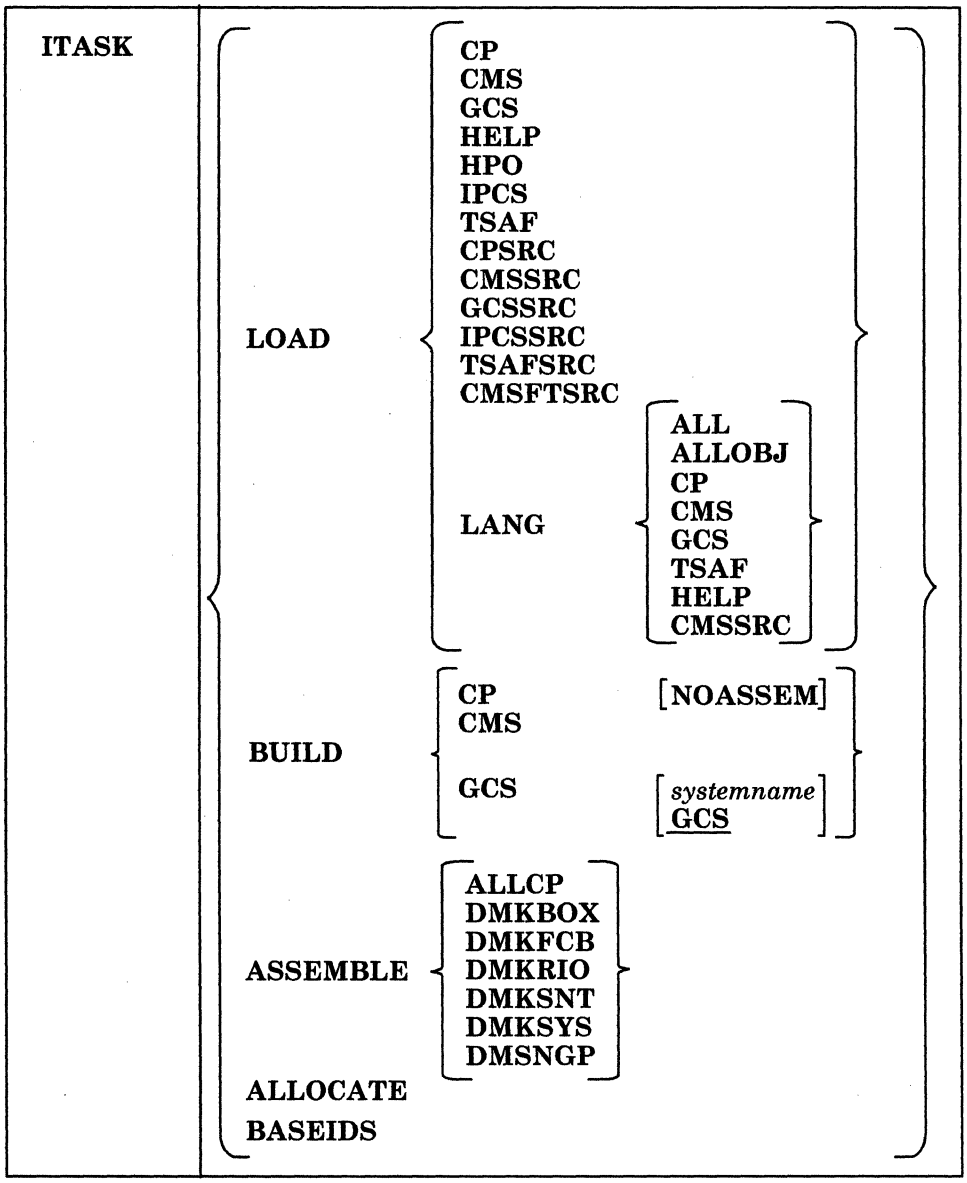
After you run GENTSAF, make sure you save the TSAF MAP file (along with the current CMS nucleus map) in case you have to process the TSAF dump at some time using IPCS. GENTSAF writes its output files to the disk accessed as A.

If you want to save any existing output file(s), save them on a disk other than the A-disk GENTSAF is writing to, or the procedure renames them.

ITASK EXEC

ITASK EXEC is a tool used primarily in the Starter System installation procedure. It invokes other EXECs and commands to perform installation and system generation tasks, which lets you to complete the installation process with fewer entries and decisions.

Command Format



LOAD

invokes SPLOAD EXEC to load files from the product tape, source feature tape, or national language feature tape to the minidisks specified in SPLOAD PROFILE. Some LOAD operands may perform additional operations.

CP

loads the following tape files from the product tape:

Tape File	Contents
SYSGEN TOOLS	SPGEN EXEC, UTILITY EXEC, SNTMAP EXEC, DISKMAP EXEC, DIRECGEN EXEC, $MSG4I$ EXEC, $DASD$ CONSTS, etc.
CP OBJECT	CP object code
SYSTEM SAMPLES	DMSNGP SAMPLE, DMKBOX ASSEMBLE, DMKFCB ASSEMBLE, RPWLIST DATA
DASDMODL SAMPLES	DIRECT *devtype[-mm]*, DMKSNT *devtype[-mm]*, EXTENTS *devtype[-mm]*, where *devtype[-mm]* identifies the DASD type and model of the VMSRES device.
DASDTYPE SAMPLES	DMKRIO *devtype*, DMKSYS *devtype*, where *devtype* identifies the DASD type of the VMSRES device.

After the files are loaded, ITASK renames the following files:

Old Name	New Name
DIRECT *devtype[-mm]*	VMUSERS DIRECT
DMKSNT *devtype[-mm]*	DMKSNT ASSEMBLE
EXTENTS *devtype[-mm]*	DIRECGEN EXTENTS
DMKRIO *devtype*	DMKRIO ASSEMBLE
DMKSYS *devtype*	DMKSYS ASSEMBLE
DMSNGP SAMPLE	DMSNGP ASSEMBLE

CMS

loads the CMS SYSTEM and CMS BASE tape files from the product tape; invokes ASMGEND EXEC to build the system assembler; invokes the SPGEN ASSEMBLE procedure to assemble DMSNGP ASSEMBLE; invokes the SPGEN BUILD procedure to build and save the CMS nucleus.

GCS

loads the GCS INTERFACE and GCS OBJECT tape files from the product tape.

HELP
loads the HELP FILES tape file from the product tape; asks you if you want to have the files converted (from mixed case, as loaded) to uppercase; issues the SAVEFD command to initialize and save the HELP file directory information in a discontiguous saved segment (DCSS) named HELP.

IPCS
loads the IPCS OBJECT tape file from the product tape.

TSAF
loads the TSAF INTERFACE and TSAF OBJECT tape files from the product tape.

CPSRC
loads the CP SOURCE tape file from the product tape.

CMSSRC
loads the CMS SOURCE tape file from the product tape.

GCSSRC
loads the GCS SOURCE tape file from the source feature tape.

TSAFSRC
loads the TSAF SOURCE tape file from the source feature tape.

CMSFTSRC
loads the CMSFT SOURCE tape file from the source feature tape.

HPO
invokes HPOINST EXEC. For more information about this EXEC, refer to the *VM/SP HPO Installation Guide.*

LANG
indicates that national language files are to be loaded from a national language feature tape.

ALL
loads the entire contents of the national language feature tape.

ALLOBJ
loads only the object (non-source) code. This consists of the following tape files: CP OBJECT, CMS BASE, GCS OBJECT, TSAF OBJECT, and HELP FILES.

CP
loads only the CP OBJECT tape file.

CMS
loads only the CMS BASE tape file.

GCS
loads only the GCS OBJECT tape file.

TSAF
loads only the TSAF OBJECT tape file.

HELP
loads only the HELP FILES tape file.

CMSSRC
loads only the CMS SOURCE tape file.

BUILD
invokes SPGEN EXEC to build the specified nucleus.

CP
accesses the MAINT 295 minidisk, which contains the CP sample files (DMKBOX, DMKFCB, DMKSNT, DMKSYS, and DMKRIO); invokes the SPGEN ASSEMBLE procedure to assemble the CP sample files; invokes the SPGEN BUILD procedure to build the CP nucleus.

If the NOASSEM option is used, the CP sample files are not assembled. This option assumes that these files have already been assembled individually.

CMS
invokes ASMGEND EXEC to build the system assembler; invokes the SPGEN ASSEMBLE procedure to assemble the CMS nucleus generation profile (DMSNGP ASSEMBLE); invokes the SPGEN BUILD procedure to build and save the CMS nucleus.

GCS
modifies a copy of the GCS loadlist (GCSLOAD EXEC) and changes the default configuration file entry (GCS) to match the filename of the GCS configuration file, *systemname* GROUP (if you do not specify a filename, the default is GCS), then writes the modified loadlist on the MAINT 295 minidisk with the other tailored files; renames the filetype of the configuration file from GROUP to ASSEMBLE; invokes the SPGEN ASSEMBLE procedure to assemble the configuration file; invokes the SPGEN BUILD procedure to build and save the GCS nucleus.

ASSEMBLE
invokes the SPGEN ASSEMBLE procedure to assemble the specified sample file.

ALLCP
assembles all of the CP sample files; DMSNGP is not assembled.

ALLOCATE
allocates space on the DASD volumes identified in the $ALLOC$ userid 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 directory; all space not specifically allocated is allocated as PERM.

BASEIDS
issues the CMS FORMAT command to format the remaining minidisks defined in the base CP directory which 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: AUTOLOG 191, CMSBATCH 195, EREP 191, OPERATNS 191 and 193, MAINT 201 and 319, CMSUSER 191, and MAINT 293, 294, 494, 496, and 596.

Messages

DMSWTK002E	*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: DMKSNT ASSEMBLE DMKRIO ASSEMBLE DMKSYS ASSEMBLE DMKBOX ASSEMBLE DMKFCB ASSEMBLE DMSNGP ASSEMBLE VMUSERS DIRECT
DMSWTK966I	Do you wish to have the HELP files converted to uppercase?
DMSWTK967R	Type: (No) or Yes
DMSWTK968I	The following minidisks defined in the base CP directory will be formatted: AUTOLOG1 191 CMSBATCH 195 CMSUSER 191 EREP 191 OPERATNS 191, 193 MAINT 201, 319 MAINT 293, 294, 494, 496, 596

DMSWTK970I	Formatting *user's vdev* minidisk
DMSWTK981R	What is the real address of your *volume* volume?
DMSWTK982R	Type: real address or SKIP
DMSWTK983E	Violation of CMS naming convention found in *args*

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

PRELOAD MODULE

The preloader is a utility program that runs under CMS. It collects multiple text files and reformats them into a single text file. The function of the preloader is similar to that of a linkage editor, but the output is in standard text file format and does not include multiple CSECTS.

A program can be developed using separate assembly modules that reference each other. The preloader can then be used to combine the assembled text files into a single loadable text file.

Command Format

PRELOAD	*loadlist* [*ctlfile*]

loadlist
specifies the filename of an EXEC on the caller's A-disk or read-only extension containing records that define input to the preloader. Each of these records contains the filename and optionally the filetype of an input text file. The format of each loadlist record defining an input file must be one of the following:

&1 &2 *fn*

&1 &2 *fn ft*

ctlfile
is the filename of an optional control file (the filetype must be CNTRL) residing on one of the caller's accessed disks. The format and interpretation of the CNTRL file is the same as that for the VMFLOAD command. It normally contains filetypes in priority sequence to be used for selecting input files if filetypes are not included in the loadlist file. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

Note: PRELOAD ignores records that have a PTF update level identifier. It then searches for the next lower level identifier to determine the filetype of the input text file. PRELOAD also ignores any options in the loadlist.

Input

The preloader gets input filenames from the loadlist. The filetype for each input file is determined in one of three ways:

1. If the loadlist record for a given input file includes a filetype entry, that filetype is used to locate the record.

2. If the loadlist record does not contain a filetype, and a control file was specified on the PRELOAD command line, the filetype constructed is in the format TXT*xxxxx*. In this case *xxxxx* is the highest control level identifier in the control file for which a file can be located on the caller's accessed disks.

3. If no filetype is specified on the loadlist entry, and a control file has not been specified on the PRELOAD command line, then the default filetype value is TEXT.

Note: Input files are located by scanning the caller's disks in their access order. All input files must be on accessed disks.

Output

The preloader output consists of two files written to the caller's A-disk:

- *fn* TEXT
- *fn* MAP.

The filename for each of these files is the same as that specified for the input loadlist file. If either of these files already exists on the caller's A-disk, the new copy replaces the old one.

TEXT File: The output TEXT file is a merged and linked composite of the input files. The first CSECT or private code section in the input expands to contain all input files. Its length is the sum of the lengths of the input files, rounded up to doubleword multiples between sections. Input TXT records of non-zero length are relocated and written to the output file.

The output RLD is a translated and relocated collection of all input RLD records. No sorting is done by the preloader. In general, each output ESD, TXT, and RLD entry appears in the same order as the corresponding input entry. ADCON and VCON fields are relocated within their TXT records. ORG statements that cause relocatable constant fields to overlay or be overlaid may cause results that differ from results obtained with a loader that completes TXT data loading prior to relocating ADCONs and VCONS.

MAP File: The output MAP file is a printable record of preloader processing, similar to a load map. The first line of the map contains:

- Output text filename
- Residence volume label and volume device address
- Date and time of file creation.

The next section of the map is a listing of the control file (if any) used. The remainder of the map contains, in processing order, a section for each input file. Each of these sections consists of:

- Filename, filetype, filemode of input file

- Residence volume label and virtual device address
- Input file's creation date and time
- Any invalid input records.

Messages

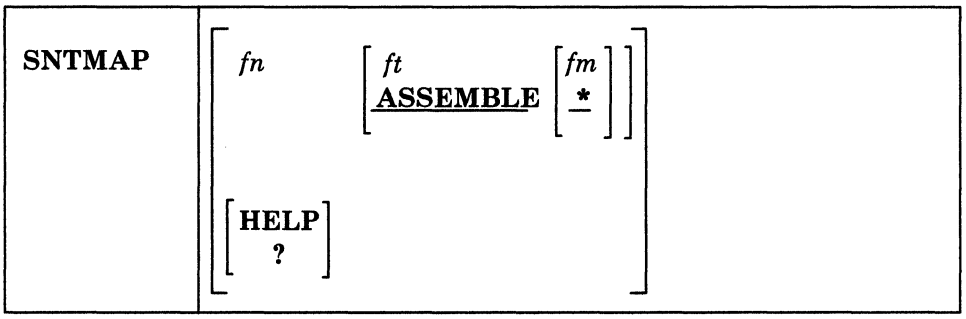
DMSPRE001E	No filename specified
DMSPRE002E	File *fn ft fm* not found
DMSPRE104S	Error *nn* reading file *fn ft fm* from disk
DMSPRE105S	Error *nn* writing file *fn ft fm* on disk
DMSPRE109S	Virtual storage capacity exceeded
DMSPRE183E	Invalid {CONTROL|AUX} file control card
DMSPRE234E	Error in LOAD LIST file *fn ft fm*[; no input]
DMSPRE235E	Error *n* in input text file *fn ft [fm]*
DMSPRE236E	Unresolved external reference(s) encountered
DMSPRE237E	Duplicate external symbol(s) encountered
DMSPRE238E	Preloader processing error

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

SNTMAP EXEC

The SNTMAP EXEC processes the macro definitions in an SNT (system name table) file and produces two CMS files: DASD SNTMAP is a saved segment DASD map; MEMORY SNTMAP is a virtual memory map.

Command Format



fn
is the filename of the SNT file. The filename must be entered, either when issuing the command or as a reply to a prompt from SNTMAP. Do not use * as the filename.

ft
is the filetype of the SNT file. Do not use * as the filetype. If you do not enter the filetype, SNTMAP assumes that the filetype is ASSEMBLE.

fm
is the filemode of the SNT file. If you do not enter the filemode, SNTMAP assumes that the filemode is *.

[HELP ?]
requests the HELP file for the SNTMAP command. This file is located on the same disk as the SNTMAP EXEC file, rather than with the other system HELP files.

Input

User Input: The SNT source file is the main input from the user. The IBM-supplied SNT source file is DMKSNT ASSEMBLE. It contains the NAMESYS, NAMENCP, NAME3800, and NAMELANG macros. SNTMAP assumes that you have assembled the file and found no syntax errors.

Note: If you want to group NAMESYS macros into families, you must insert comment lines into the SNT file. This is explained in the **VM/SP Planning Guide and Reference**.

The CP directory file is another user input to SNTMAP. The directory contains a USER $SAVSYS$ entry that describes the DASD areas allocated

for SNT data. Within USER $SAVESYS$, the MDISK statements must be listed in ascending order by cylinder/block number.

After you invoke SNTMAP, the EXEC prompts you to enter the name of the directory. If SNTMAP cannot find a USER $SAVSYS$ entry, or if one or more MDISK statements define the DASD type as FB-512, SNTMAP will prompt you to enter the DASD types. In this case, SNTMAP will not check for $SAVSYS$ extent exceeded conditions.

Other Input: SNTMAP also uses the file $DASD$ CONSTS, which is supplied with VM/SP. This file contains information about DASD that SNTMAP needs for calculations and conversions. SNTMAP cannot do any processing without this file.

Output

SNTMAP creates two CMS files:

- DASD SNTMAP
- MEMORY SNTMAP

that you can display at a terminal, edit (using XEDIT), and print. When these files are created, any old files by the same name are erased. Samples of these files are shown in the *VM/SP Planning Guide and Reference.*

DASD SNTMAP: This file contains a list of the DCSS names and DASD allocations for each pack that has DCSS resources defined. The list is arranged in order of occurrence on a given pack. Any gaps are noted within the list and summarized in a table at the end of the file.

MEMORY SNTMAP: This file is a list of DCSS names and segment allocations arranged in order by memory location. Overlapping segments are noted within the list. If you have grouped NAMESYS macros into families, this file will show a separate memory map for each family. This file maps only the shared segments defined by a NAMESYS macro. Unshared pages are listed at the end of the file.

Messages

DMSWSM006E	NO READ/WRITE 'A' DISK ACCESSED
DMSWSM847R	Enter name of SNT definition file or press ENTER to quit.
DMSWSM848E	Unable to proceed without '$DASD$ CONSTS' file information.
DMSWSM849R	Enter '*volid*' DASD type or type 'QUIT' to end SNTMAP processing.
DMSWSM850W	'*devtype*' is not a valid DASD type.
DMSWSM851W	Page number exceeds device limit of *limit* for '*sysname*'. SYSSTRT parameter = *parameter*
DMSWSM852E	SYSSTRT parameter for '*sysname*' is not compatible with *devtype* DASD type for *volid*. SYSSTRT parameter = *parameter*

DMSWSM853R	Enter the name of your current CP directory file Default is 'VMUSERS DIRECT'.
DMSWSM854W	Unable to find CP directory file: *fn ft fm*. No $SAVSYS$ boundary checking will be done.
DMSWSM855W	No $SAVSYS$ areas defined in the CP directory file. No $SAVSYS$ boundary checking will be done.
DMSWSM884I	Results of mapping are in two CMS files: DASD related information is in 'DASD SNTMAP'. Memory related information is in 'MEMORY SNTMAP'.

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

SPGEN EXEC

The SPGEN EXEC performs various system generation and maintenance functions, using the information contained in SPGEN PROFILE (or a specified profile). These functions include:

- Creating, verifying, and displaying system profile parameters.
- Assembling system files.
- Generating CP, CMS, and GCS nuclei.
- Receiving and verifying load maps.

Command Format

SPGEN

Create [*(optionA[ )]]

Verify [*(optionA[ )]]

Display [*compname* / **ALL**] [*(optionA[ )]]

Assemble *fn* [*compname* / **CP**] [*(optionA[ )]]

Setup *compname* [*(optionA[ )]]

Nucleus *compname* [*(optionA optionB optionC [ )]]

Map *compname* [*(optionA[ )]]

DTYPE *vdev*

Options:

optionA: [**PROFile** *profname*]
optionB: [**NOIPL**]
optionC: [**NOSETUP**]

Create
writes the SPGEN PROFILE (or the specified profile) on your A-disk.

Verify
reads and parses the SPGEN PROFILE (or the specified profile), displaying messages if errors are encountered in the file.

Display
displays profile keyword values and their associated commentary from SPGEN PROFILE (or the specified profile) for ALL components (or the specified component).

compname
is a valid component name. This can be a standard IBM component name (CP, CMS, IPCS, GCS, TSAF) or, if the profile has been properly updated, a user-created name.

Assemble *fn*
performs system assemblies by processing the SPGEN PROFILE (or the specified profile) and then invoking VMFASM EXEC for the specified filename (the filetype is not specified, but must be ASSEMBLE), using the control file for CP (or the specified component) indicated in the profile.

Setup *compname*
runs each instruction in the list that follows the *compname*_setup tag in SPGEN PROFILE (or the specified profile). SETUP instructions may be any combination of ACCESS, LINK, or EXEC commands.

Nucleus *compname*
builds a system nucleus for the specified component as follows:

1. Executes the list of instructions that follows the *compname*_setup tag in SPGEN PROFILE (or the specified profile), unless the NOSETUP option is selected.
2. Spools the punch as class N (for Nucleus) and spools the printer as class M (for Map).
3. Issues the VMFLOAD command, using the loadlist and control file names from SPLOAD PROFILE (or the specified profile) as parameters.
4. After VMFLOAD is complete, spools the reader as class N and IPLs the reader to generate the nucleus, unless the NOIPL option is specified.

Map *compname*
names the first available class M reader file to whatever file specification is indicated by the *compname*_mapname tag in SPGEN PROFILE (or the specified profile). If the *compname*_mapuserid tag in the profile indicates a userid that is different from the one that invoked the SPGEN MAP function, the load map is transferred to the userid indicated in the profile; otherwise, the load map is left in the reader.

DTYPE *vdev*
issues a DIAGNOSE code X'24' for the specified virtual device address, uses the returned information to look up the DASD type in the $DASD$ CONSTS file, and returns that information to the terminal (if SPGEN DTYPE is executed) or to the program stack (if SPGEN DTYPE is called).

Only VM/SP-installation-supported DASD types are represented in $DASD$ CONSTS. The information returned is a single line consisting of three fields:

- DASD type and model
- Capacity in cylinders or FB-512 blocks
- The proper representation of the given DASD model to the FORMAT/ALLOCATE program.

If the virtual device address is invalid, non-existent, or addresses an unsupported device type, a non-zero return code is displayed.

The SPGEN DTYPE function is intended primarily for use by installation-related utilities.

options...

PROFile *profname*
indicates that a profile with the specified filename (the filetype is not specified, but must be PROFILE) should be used instead of SPGEN PROFILE.

NOIPL
prevents the reader from being IPLed for a nucleus build. This enables later transfer of the reader image to tape, for example.

NOSETUP
bypasses the execution of any commands in the *compname*_setup section of the profile.

SPGEN PROFILE Syntax

SPGEN EXEC uses information supplied by a profile. SPGEN PROFILE is the default profile (see "Sample SPGEN PROFILE" on page C-3). The profile, which must have a filetype of PROFILE, contains lines that are either comments (beginning with /*), keywords and values, or SETUP tag commands. Blank lines are also allowed and are treated as comment lines.

A keyword begins with a colon and ends with a period, and may not contain imbedded blanks. The keyword consists of a component identifier, an underscore character, and a tag. The permitted keywords, tags, and values are shown below:

```
[/* [general comment imbedded in the profile (not used)]]
:CP_small.          { no }                          [/*[comment]]
                    { yes }
                    { * }
:CP_processor.      { UP }                          [/*[comment]]
                    { AP }
                    { MP }
                    { * }
:CP_fret.           { no }                          [/*[comment]]
                    { yes }
                    { * }
:CP_vr.             { no }                          [/*[comment]]
                    { yes }
                    { * }

:compname_control.  { fn }                          [/*[comment]]
                    { * }
:compname_loadlist. { fn }                          [/*[comment]]
                    { * }
:compname_lang.     { langid }                      [/*[comment]]
                    { AMENG }
                    { * }
:compname_mapname.  { fn [ ft ] }                   [/*[comment]]
                    { compnameNUC [ MAP ] }
                    { * }
:compname_mapuserid. { userid }                     [/*[comment]]
                     { MAINT }
                     { * }
:compname_setup.                                    [/*[comment]]

  [(LINK, ACCESS, EXEC commands)]                   [/*[comment]]
```

:CP_small.
indicates whether or not the Small CP option is selected. The default is no. This information is used only if the CP loadlist name is not explicitly stated in the profile.

:CP_processor.
indicates whether a UP, AP, or MP system is being built. The default is UP. This information is used only if the CP control file and loadlist names are not explicitly stated in the profile.

:CP_fret.
indicates whether or not the CP FRET Trap option is selected. The default is no. This information is used only if the CP control file name is not explicitly stated in the profile.

:CP_vr.
indicates whether or not a Virtual=Real system is to be built. The default is no. This information is used only if the CP loadlist name is not explicitly stated in the profile.

:*compname*_control.
indicates the control file name associated with the specified component. The filetype is not specified, but must be CNTRL.

:*compname*_loadlist.
indicates the loadlist filename associated with the specified component. The filetype is not specified, but must be EXEC.

:*compname*_lang.
indicates the langid of the system national language to be used with the specified component. The default langid is AMENG (American English).

:*compname*_mapname.
indicates the filename and filetype to be assigned to the load map associated with a nucleus generation for the specified component. The default filename is *compname*NUC; the default filetype is MAP.

:*compname*_mapuserid.
indicates the virtual machine userid to which a load map is transferred for the SPGEN MAP function. The default for the CP, CMS, and GCS components is MAINT. SPGEN transfers the map after the map has been named according to the specification in the mapname tag described previously.

:*compname*_setup.
is a keyword that announces that the following lines are ACCESS, LINK, or EXEC commands, to be invoked when the SETUP and NUCLEUS functions are used. The list is terminated either by the occurrence of another valid keyword or by the end of the profile file.

No values may be listed on the same line as the setup tag, and any comments on the line are ignored. Comments on command lines following a setup tag are displayed with the commands when the DISPLAY function is invoked.

compname
is a valid component name. It may be a standard IBM component name (such as CP, CMS, GCS, TSAF), or it may be a user-defined component name. A user-created component name can be any string of alphanumeric characters, the plus sign (+), and the hyphen (-), whose length when combined with the keyword commentary does not exceed 200 characters.

*
indicates that the default value is to be used (see the next section).

comment
is any commentary that may be helpful in describing keyword entries in the DISPLAY function. A comment line should not exceed 200 characters. Only the first 35 characters of the comment are used in the DISPLAY function output.

SPGEN PROFILE Defaults

The following defaults are used when an asterisk (*) is entered as the value for a keyword in the profile.

The default CP control file name depends on the values specified in the profile for CP_fret (no or yes) and CP_processor (UP, AP, or MP), as follows:

	UP	AP	MP
fret no	DMKSP	DMKSPA	DMKSPM
fret yes	DMKSPT	DMKSPAT	DMKSPMT

The default CP loadlist name when using the CP_small option is CPLOADSM. This loadlist does not include V=R, AP, or MP support.

For any nucleus other than Small CP, the loadlist name depends on the values specified in the profile for CP_vr (no or yes) and CP_processor (UP, AP, or MP), as follows:

	UP	AP	MP
vr no	CPLOAD	APLOAD	APLOAD
vr yes	VRLOAD	AVLOAD	AVLOAD

The remaining CP keywords have default values as follows:

Keyword	Default
CP_small	NO
CP_processor	UP
CP_fret	NO

Keyword	Default
CP_vr	NO
CP_lang	AMENG
CP_mapname	CPNUC MAP
CP_mapuserid	MAINT

Default values for the remaining components are as follows:

	CMS	GCS	HPO
Control File	DMSSP	CSISP	
Loadlist	CMSLOAD	GCSLOAD	
Language	AMENG	AMENG	AMENG
Map Name	CMSNUC MAP	GCSNUC MAP	CPHPONUC MAP
Map Userid	MAINT	MAINT	MAINT

For information about the contents of the default control files, see Appendix L, "Control File Identifiers" on page L-1.

Usage Notes

1. SPGEN EXEC always sets EMSG to ON before performing the desired function. This ensures that complete message numbers and messages are displayed if errors occur. The original EMSG setting is restored once the function is completed.

2. All keyword, component, file specification, and tag names must consist only of characters that are valid for CMS file specifications: the alphanumeric characters, the plus sign (+), the hyphen (-), the colon (:), and the underscore (_). No imbedded blanks are allowed.

3. Multiple systems may be maintained in two different ways using SPGEN. Individual profiles may be set up for each system, or a single profile may be set up with sets of keywords for each system.

   In the latter case, the systems are identified by the component portion of the keywords. For example, different CP loadlist names could be specified as CP1_loadlist and CP2_loadlist. Similarly, each system may have different minidisk access orders defined by corresponding entries under CP1_setup and CP2_setup keywords.

4. Nucleus builds leave the printer spooled as class M, the punch and reader spooled as class N, and both printer and punch spooled to the reader.

5. Any valid ACCESS, LINK, or EXEC commands may be listed under the setup tags in the profile. When SPGEN invokes these commands, it checks the return code of each one, and will not proceed with any other operations if it encounters any non-zero return codes.

Before any ACCESS command is invoked, SPGEN first releases the disk to be ACCESSed to insure that the ACCESS command has the opportunity to send a zero return code, even if the disk was previously ACCESSed as a R/W minidisk.

6. For CP, CMS, and GCS entries in the profile, default entries may be selected by using an asterisk (*) as the keyword value, except in the SETUP tags. The CP_loadlist and CP_control keywords are specially processed by SPGEN if an asterisk is used as the keyword value to indicate a default selection. In this case, other CP keyword values are used to determine which of the default CP loadlist and control files should be used.

7. Invoke the SPGEN DISPLAY function to examine the values that will be used for the various operations.

8. A slash asterisk (/*) identifies a comment in the profile file. Anything following these characters on a non-keyword line is ignored. The first 35 characters of any comment on a keyword value or on a command entry under a setup tag are saved and used for the SPGEN DISPLAY function.

9. SPGEN will not continue processing if it finds errors or inconsistencies when parsing the profile file. This is true even if these errors occur in keywords that have nothing to do with the specified function.

10. If the CP loadlist and control file keywords indicate default values (with *), SPGEN stops processing if it finds a combination of CP_small, CP_processor, CP_fret, and CP_vr keyword values that is invalid.

    For example, if the CP_small keyword value is yes and the CP_processor keyword value is MP, an error message is issued to indicate incompatible options, since no loadlist file is supplied to build a nucleus with these characteristics.

    However, if the CP_loadlist keyword is assigned a specific loadlist file specification, SPGEN ignores the inconsistencies in the keyword values mentioned above.

11. SPGEN is not dependent on the case (upper or lower) of any alphabetic characters in a profile.

Messages

DMSWSG003E Invalid option: *option*
DMSWSG014E Invalid keyword *keyword*
DMSWSG024E File *fn ft fm* already exists
DMSWSG026E Invalid value *value* for *keyword* keyword
DMSWSG032E Invalid filetype *ft*
DMSWSG047E No function specified
DMSWSG050E Parameter missing after *value*
DMSWSG065E *option* option specified twice
DMSWSG069E Disk *mode*[(*vdev*)] not accessed

DMSWSG070E Invalid parameter *parameter*
DMSWSG095E Invalid address *vstor*
DMSWSG252E Invalid filename *fn*
DMSWSG637E Missing value for the *option* option
DMSWSG643E No class *fileclass* files in your reader
DMSWSG957E Incompatible *option* options for *options*
DMSWSG958I No *keyword* initialized in *fn ft fm*
DMSWSG959E Missing value for *keyword* keyword
DMSWSG960E Error in line *line* of *fn ft fm*
DMSWSG961E There are class *class* files in the *device*
DMSWSG962E Invalid character: *character*
DMSWSG984E Too many class *class* files in *device*
DMSWSG985E Conflicting values for duplicate keyword *keyword*

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

SPLOAD EXEC

The SPLOAD EXEC loads files from the VM/SP Product Tape, source feature tape, and national language feature tape according to the load instructions contained in SPLOAD PROFILE.

Command Format

SPLOAD	*group*	*element*	[*fn* [*ft*]]
			[*] [*]

group element
is a tape file identifier that SPLOAD EXEC uses to locate the entry in SPLOAD PROFILE that contains the load instructions for this tape file.

fn [*ft*]
form a file specification template. By default, both are set to asterisks (*), meaning all files within a tape file. However, they may be set to a specific filename and/or filetype to selectively load specific files from a tape file to a minidisk.

SPLOAD PROFILE Syntax

In the sample SPLOAD PROFILE, data entries are organized by format number (see definition below), and format number sets are separated by comments lines (see "Sample SPLOAD PROFILE" on page C-1). A comment line begins with a slash asterisk (/*). Each data entry in SPLOAD PROFILE has the following syntax:

```
group  element  userid  address  format  volume  fileno
```

group element
is a tape file identifier that identifies the profile entry containing the load instructions for this tape file.

userid
is the owner of the minidisk to which the tape file contents are loaded. This information is used only if SPLOAD EXEC must LINK and ACCESS the minidisk in order to load the file.

address
is the minidisk address to which the tape file is loaded.

format
is the tape format. 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.

SPLOAD EXEC selects the appropriate entries in the profile depending on the user's tape format.

volume
is the product tape volume on which the particular tape file resides.

fileno
is the relative tape file number (location) of the specified tape file on the particular product 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 SPLOAD EXEC uses to determine which tape is mounted, and therefore which entries to use from SPLOAD PROFILE. SPLOAD EXEC loads $TAPE$ HEADER to the user's A-disk each time SPLOAD is invoked. If the user does not have a read/write A-disk, SPLOAD terminates with a return code of 36.

2. SPLOAD EXEC expects that the tape drive being used is ATTACHed at virtual address 181. If this condition is not met, SPLOAD can not continue.

3. SPLOAD EXEC ACCESSes the target minidisk as "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 the target minidisk is not LINKed, SPLOAD attempts the LINK. However, if a password is required, SPLOAD terminates.

6. If the 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.

7. 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 userid which owns the minidisk (except MAINT, which uses the characters MNT) followed by the three-digit minidisk address.

8. SPLOAD EXEC restores the minidisk access order to its original state just before exiting. However, since SPLOAD performs only standard ACCESSes, the ACCESS that is restored for a given minidisk may be different from the original ACCESS if that disk 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 *user vdev*
DMSWSL737R	Enter the minidisk address for the *group element*
DMSWSL963E	*keyword value* not found in *fn ft fm*
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 *mdisk*

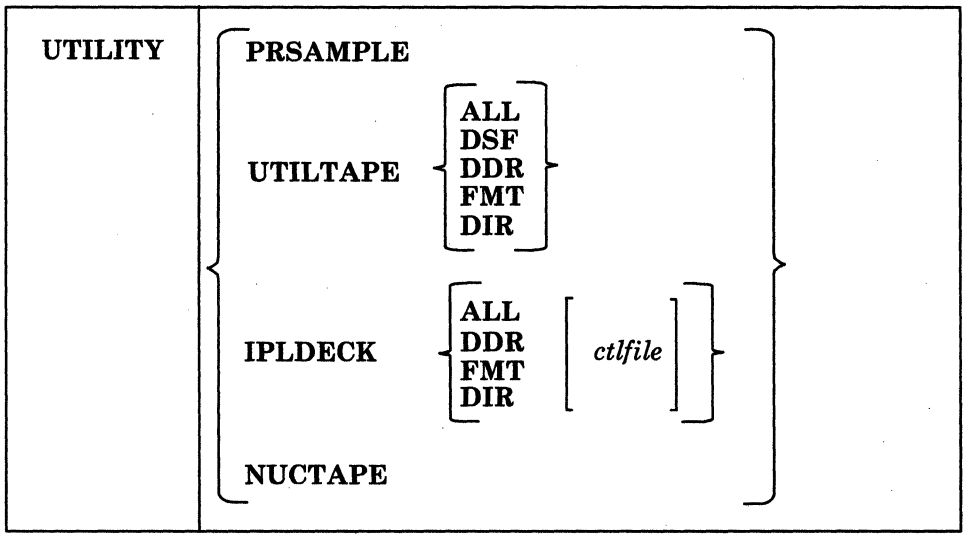
For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

UTILITY EXEC

The UTILITY EXEC provides occasionally-used installation utility functions:

- Printing the system definition files
- Creating a stand-alone service utility tape
- Writing a backup IPLable copy of the CP nucleus to tape
- Creating stand-alone service programs on disk.

Command Format



PRSAMPLE
prints the following system definition files:

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

These files are spooled to the system printer. The minidisk(s) containing these files must be accessed before using the PRSAMPLE operand.

UTILTAPE
creates a tape containing any or all of the following stand-alone utility programs:

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

Each IPLable program is written as a separate tape file, with one or more tape marks separating the programs. Therefore, when you IPL a program off the tape, you may have to IPL more than once to bypass the tape marks.

IPLDECK
creates any or all of the following stand-alone utility programs on disk from their associated object modules (text decks):

- DIR
- FMT
- DDR.

You are prompted for the target disk (which is accessed as filemode A). Make sure that the minidisk containing your latest level TEXT or TXT* decks is accessed.

ctlfile
is the name of an optional control file (the filetype must be CNTRL) that determines the filetype of the text deck used to create the stand-alone service program. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

NUCTAPE
writes a backup IPLable copy of the CP nucleus to tape.

Messages

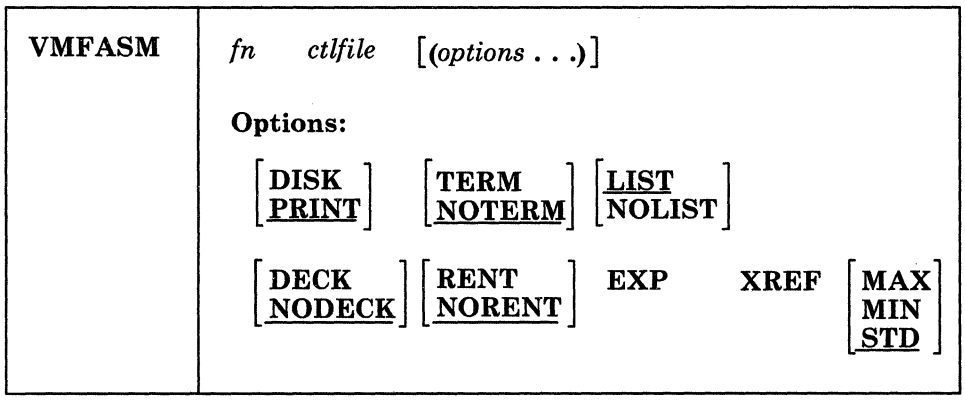
DMSWUT050E	Parameter missing after *value*
DMSWUT070E	Invalid parameter *parameter*
DMSWUT095E	Invalid address *vstor*
DMSWUT971E	Unable to locate *fn ft*
DMSWUT972I	*fn ft fm* spooled to printer
DMSWUT973R	Enter the minidisk address where the IPL decks were loaded
DMSWUT974I	Unable to find IPL decks on the minidisk you indicated
DMSWUT975I	Moving *fn ft* to tape
DMSWUT976I	The *fn ft* program is on tape file number *number*
DMSWUT977I	Enter disk address where the service programs are to be written
DMSWUT978R	Type: (190) or cuu
DMSWUT979I	IPL *vdev* has been created
DMSWUT980I	An IPLable CP nucleus now exists on tape
DMSWUT986I	Unable to restore ACCESS to *mdisk*

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

VMFASM EXEC

The VMFASM EXEC procedure updates a specified source file according to entries in a control file and then assembles the updated source file. VMFASM invokes the CMS UPDATE and ASSEMBLE commands.

Command Format



fn
is the filename of the source file to be updated. The filetype must be ASSEMBLE.

ctlfile
is the filename of the control file. The filetype must be CNTRL. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

options...
VMFASM accepts only the following assembler options. All other assembler options entered are ignored and the defaults are used.

DISK
places the listing file on a virtual disk.

PRINT
writes the listing file to the printer.

TERM
writes the diagnostic information on the SYSTERM data set. The diagnostic information consists of the diagnosed statement followed by the error message issued.

NOTERM
suppresses the TERM option.

LIST
produces an assembler listing.

NOLIST
does not produce an assembler listing.

DECK
writes an object module on the device specified on the FILEDEF PUNCH statement.

NODECK
suppresses the DECK option.

RENT
checks the program for a possible violation of program reenterability. Code that makes the program nonreenterable is identified by an error message.

NORENT
suppresses the RENT option.

EXP
expands printing of certain macros which check for the SUP parameter issued via the SYSPARM option of the ASSEMBLE command. The default is SUP.

XREF
invokes the XREF(FULL) assembler option. The default for VMFASM is XREF(SHORT).

MAX
invokes the BUFSIZE(MAX) assembler option. If both MAX and MIN are specified, then MIN takes precedence.

MIN
invokes the BUFSIZE(MIN) assembler option. If both MAX and MIN are specified, then MIN takes precedence.

STD
invokes the BUFSIZE(STD) assembler option. This is the default for VMFASM when neither MAX nor MIN is selected.

How VMFASM Works

The steps taken by the VMFASM EXEC are summarized below.

1. The VMFASM EXEC calls the UPDATE command with the CTL, STK, PRINT, and OUTMODE A1 options.

   UPDATE uses the control file (*ctlfile* CNTRL) to update the assembler language source file. If updates exist, the new file is named *$fn* ASSEMBLE. Otherwise, it is called *fn* ASSEMBLE.

   UPDATE stacks information from the control file in the console stack and prints the update log file.

The OUTMODE A1 option specifies that the files created by the UPDATE command will be written onto the A-disk with a filemode of A1.

If a PTF (program temporary fix) file is missing, a message stating this will be written to both the console and to the update log file.

2. Using the library list from the MACS record in the control file, VMFASM issues a GLOBAL MACLIB command.

3. The updated source file, $*fn* ASSEMBLE, is assembled using the options indicated on the VMFASM command line.

4. The output text deck from the assembly, $*fn* TEXT, is concatenated with the UPDATES file so that the text deck contains a history of update activity.

5. Using the update level identifier from the control file (the identifier of the most recent update that was found and applied is stacked by the UPDATE command), VMFASM determines how to rename $*fn* TEXT.

   If the update level identifier is TEXT, the text deck is renamed *fn* TEXT.

   If the update level identifier is anything other than TEXT, the text deck is renamed *fn* TXT*xxxxx* (where *xxxxx* is the 1-to-5-character update level identifier). The new *fn* TEXT (or TXT*xxxxx*) is found on the A-disk.

   Note: The new TEXT deck will be filemode A1 regardless of the filemode of the original TEXT deck. If the filemode of the original TEXT deck is filemode 2, the TEXT deck created by VMFASM will have to be renamed to filemode 2.

6. Temporary files $*fn* ASSEMBLE (or *fn* ASSEMBLE), *fn* UPDATES, and *fn ctlfile* are erased.

Input and Ouput Files

Disk Input Files: VMFASM uses the following disk input files:

fn ASSEMBLE — Assembler language source file

ctlfile CNTRL — Control file

Other input may include MACLIBs, auxiliary control files (*fn* AUX*xxxxx*), and miscellaneous update files.

Disk Output File: VMFASM creates the following disk output file:

fn { TEXT / TXT*xxxxx* } — Object deck, named according to the update level identifier in the control file

This file also contains data from the UPDATES file, together with date and time information.

Printer Output File: If the PRINT option is in effect, VMFASM creates the following printer output file:

fn ctlfile	Assembler listing

This file also contains data from the update log file (*fn* UPDLOG), describing the updates applied to the source file.

Responses

fn TEXT A HAS BEEN ERASED.

The text file to be produced does not have a filename of TEXT. File '*fn* TEXT A' already existed. This prevents you from inadvertently including a down-level textfile in your nucleus.

ASMBLING fn

The assembly is going to begin. If you specified any assembler options on the VMFASM command line, the options used are also displayed.

fn { **TEXT** / **TXTxxxxx** } **A1 CREATED**

indicates the filename and filetype of the text deck.

***** ERROR UPDATING fn *****

A return code from the UPDATE command was greater than 12 but not equal to 40. This means that a severe error occurred. VMFASM ends with the return code from the UPDATE command.

***** ERROR ASMBLING fn *****

An assembler error occurred.

***** fn** { **TEXT** / **TXTxxxxx** } **A1 IS NOT CREATED *****

No text file was produced because of assembler errors. VMFASM ends with the return code from the ASSEMBLE command.

fn { **TEXT** / **TXTxxxxx** } **A WAS NOT CREATED, RC=rc FROM COPYFILE**

The COPYFILE command failed to merge the *fn* UPDATES A1 files, or your A-disk is full. Therefore, the text file failed with the return code shown. VMFASM ends with the return code from the COPYFILE command.

Messages

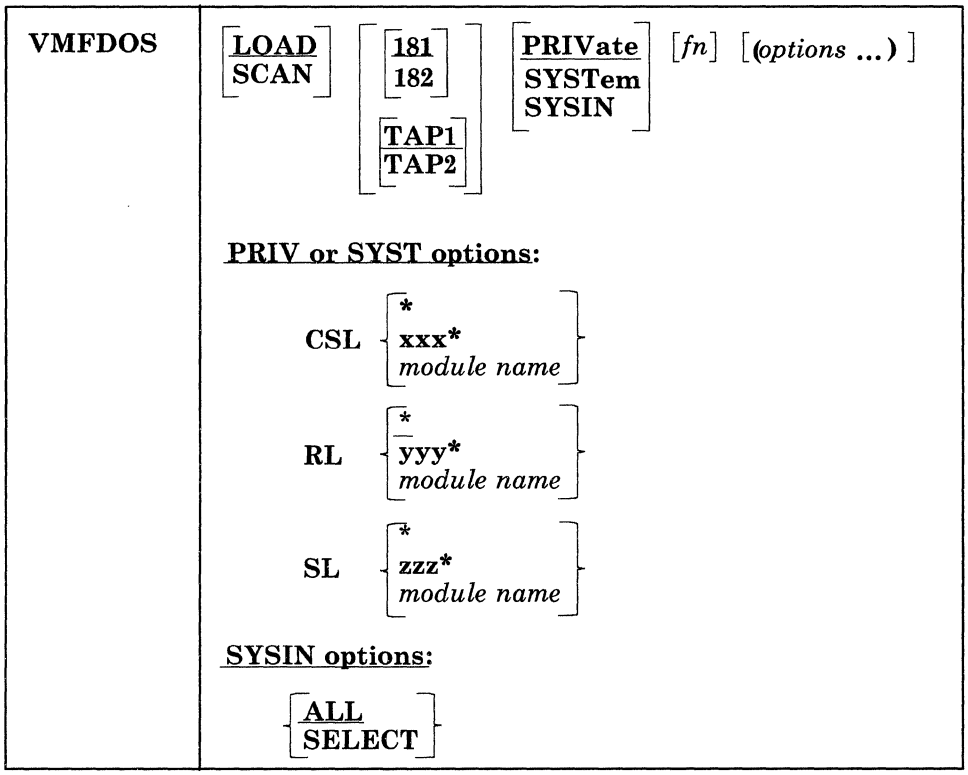
DMSSTT002E File *fn* ASSEMBLE not found
DMSSTT002E File *ctlfile* CNTRL not found
DMSUPD178I (Applying|Updating) *fn ft fm*
DMSUPD181E No update files were found

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

VMFDOS MODULE

The VMFDOS command creates CMS files containing VSE modules 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.

Command Format



LOAD
: loads the specified VSE modules as CMS files from the VSE tape. The modules to be loaded are specified:

- Via the SL or RL options
- Via 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.

{TAP*n* / 18*n*}
specifies the symbolic tape identification (TAP*n*) or the actual device address (18*n*) of the tape. TAP1 or 181 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 filetype 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 filename to create a file called *fn* SCAN that lists the names of all the VSE modules contained on the tape.

options...
The following options may 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 filename 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.

options...
The following options may 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.

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 filetype of 'TEXT'; they have a filename 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 filetype of 'TEXT'; they have a filename 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 filetype of 'MACRO'; they have a filename 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 filetype of 'MACRO'; they have a filename 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 filename the same as the VSE module name and a filetype of 'TEXT' or 'MACRO', as appropriate.

vmfdos scan syst

The VSE system libraries are scanned. File 'VMFDOS SCAN' is created to contain the names of the VSE modules.

vmfdos scan sysin fn

The VSE SYSIN tape is scanned. File 'fn SCAN' 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 filename the same as the VSE module name and a filetype of 'TEXT'.

vmfdos sysin fn

File 'fn SCAN' 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 filename the same as the VSE module name and a filetype of 'TEXT' or 'MACRO', as appropriate.

vmfdos sysin

File 'VMFDOS SCAN' 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 filename the same as the VSE module name and a filetype of 'TEXT' or 'MACRO', as appropriate.

Usage Notes

1. When the scan function is invoked, VMFDOS erases either the CMS file 'fn SCAN A1' or 'VMFDOS SCAN A1' prior to scanning the tape.

2. The tape must be positioned at the file to be processed (via the TAPE FSF command) prior to issuing the VMFDOS command.

3. VMFDOS erases those existing files that have the same filename as a VSE module being loaded.

4. VMFDOS uses the CMS 'A' disk to write all files. A message is issued if a R/W 'A' disk is not available.

5. VSE modules loaded from a relocatable library tape file will have a filetype of 'TEXT' when written as a CMS file.

6. VSE modules loaded from a source statement library tape file will have a filetype of 'MACRO' when written as a CMS file.

7. If a filename is specified when using the LOAD function, any options specified are ignored and the 'fn SCAN' file is used. If a filename 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.

Messages

DMSDOS787E ERROR DECOMPRESSING *macro* MACRO ONLY THE FIRST 430 ENTRIES LISTED WILL BE PROCESSED, BUFFER IS FULL

An informational message only; processing continues. VMFDOS only acquires enough storage to contain 430 entry requests when a filename is specified or 'VMFDOS SCAN' exists and is used as input.

VMFLKED EXEC

The VMFLKED EXEC procedure invokes the CMS LKED command to link-edit modules into a LOADLIB. VMFLKED uses the normal CMS search order when searching for TEXT files.

Command Format

VMFLKED	*fn* [*ft* / **LKEDCTRL** [*fm* / *]] [(*options*...[)]]

fn
is the filename of the input control file. You must specify a filename.

ft
is the filetype of the input control file. The default filetype is LKEDCTRL.

fm
is the filemode of the input control file. The default filemode is *.

options...

PRINT
prints out a hardcopy of the linkage editor output.

MODULE *module_name*
indicates that only those members of the LOADLIB that include *module_name* are to be link-edited.

How VMFLKED Works

VMFLKED reads the specified LKEDCTRL file and expects to find option records (if any are needed) followed by linkage editor input records. Multiple groups of options followed by linkage editor input can be combined in a single LKEDCTRL file (see Figure 15-2 on page 15-52).

1. VMFLKED processes the option records, which are identified by '%' in column one.

2. VMFLKED processes linkage editor input records when it finds a non-option and non-commentary record.

   When VMFLKED finds:

   - an INCLUDE record, then the process adds the record to the linkage editor input file and issues a FILEDEF for the TEXT file.

- a NAME record, then the process adds the record to the linkage editor input file and invokes the linkage editor.
- a commentary record (beginning with *), then the process ignores the record.
- any other record, then the process adds the record to the linkage editor input file.

3. VMFLKED continues with Step 1 to process the next group of option records (if any).

Input and Ouput Files

Input Files

fn TEXT	the name of the TEXT file. When the process finds an INCLUDE statement in the linkage editor input control file, it issues a FILEDEF for that TEXT file. The linkage editor reads this TEXT file.
fn LKEDCTRL	a modified linkage editor file. INCLUDE cards include TEXT files, and you cannot use them to include files from a library. Any record containing an asterisk (*) in column one is commentary and the process ignores it. The VMFLKED EXEC also recognizes special control (option) records in the linkage editor control file (see Figure 15-1 on page 15-50). These records, which must begin with a '%' in column one, may only be located between linkage editor input files (i.e., groups of them may be at the beginning of the file or following a NAME record).

Record	Parameter	Function
%CONTROL	control_file_name	This record indicates to take the filetypes for linkage editor input files from a control file. The name of the control file is specified on the %CONTROL record and the filetype is CNTRL. When the %CONTROL record is read, VMFLKED reads the control file and uses the first "word" of each line of the file to build an array of filetypes. The first line of the file indicates the default filetype. Each of the filetypes is checked and if it is not TEXT, then the characters "TXT" are put at the beginning. When an VMFLKED finds an INCLUDE card, it searches the array. VMFLKED uses the first filetype from the array which matches an existing file. This option is in effect until a %NOCONTROL record is found.
%NOCONTROL		This record indicates not to take the filetype for linkage editor input files from a control file.
%LIBRARY	load_library_name	This record changes the LOADLIB to be used and the name on the LKEDIT listing. The default for the LOADLIB name is the filename of the LKEDCTRL file.
%LEPARMS	link-edit parameters	This record sets the link-edit parameters used in each link-edit step. If no parameters are specified, then none are used.
%MAXRC	maximum_valid_return_code	This record sets the maximum valid return code. The VMFLKED EXEC checks this value when it ends. If the highest return code is higher than this value and is not listed on the %ACCEPTRC record, then the EXEC issues a warning message.
%ACCEPTRC	return code(s)	This record lists the acceptable return code(s) for VMFLKED processing. (The return code(s) are usually higher than entry on the %MAXRC entry). VMFLKED checks the value(s) after each link-edit. If the return code(s) after any link-edit is higher than the %MAXRC record and not specified on the $ACCEPTRC record, then VMFLKED issues a warning message at the end of its processing.
%IGNORE		This record causes the EXEC to bypass the warning message for missing TEXT files. Once specified, this record takes effect for all subsequent link-edits (or until a %NOIGNORE record is found).

Figure 15-1 (Part 1 of 2). Linkage Editor Control File Special Control (Option) Records

Record	Parameter	Function
%NOIGNORE		This record causes a warning message to be issued if there is a missing TEXT file. Once specified, this record takes effect for all subsequent link-edits (or until a %IGNORE record is found).
%ERASE		This record erases LOADLIB and LKEDIT files. This is useful when you rebuild the LOADLIB as it keeps the LOADLIB and LKEDIT files as small as possible. If this record is not specified, then the LOADLIB and LKEDIT files are not erased.

Figure 15-1 (Part 2 of 2). Linkage Editor Control File Special Control (Option) Records

Note: The %ERASE control statement takes effect immediately, and erases whatever is the current LOADLIB. The LOADLIB is not erased if you use the MODULE option. (The LOADLIB can be changed using the %LIBRARY control statement).

Output Files

fn LOADLIB — the main output from the linkage editor. This file contains the link-edited load modules.

fn LKEDIT — the file which contains the linkage editor map for all modules

The following is an example of a LKEDCTRL file:

```
%CONTROL YOURCTRL
%LIBRARY NCCF
%ERASE
%MAXRC 4
%ACCEPTRC 12 14
%LEPARMS NCAL LIST XREF LET RENT
  INCLUDE  DSIZDST
  INCLUDE  DSIZSHP
  INCLUDE  DSILUTRM
  ORDER    DSIZDST
  ENTRY    DSIZDST
  NAME     DSIZDST(R)
%LEPARMS NCAL LIST XREF LET REUS
  INCLUDE  DSIVMCMD
  INCLUDE  DSIVMMSG
  ORDER    DSIVMCMD
  ENTRY    DSIVMCMD
  NAME     DSIVMCMD(R)
%IGNORE
%LEPARMS NCAL LIST XREF LET RENT
  INCLUDE  DSIPRTVM
  ENTRY    DSIPRTVM
  NAME     DSIPRTVM(R)
```

Figure 15-2. Example of a LKEDCTRL file.

Messages

DMSWLK002E	File *fn ft [fm]* not found
DMSWLK002W	File *fn ft* not found
DMSWLK003E	Invalid option *opt*
DMSWLK005E	No {LKEDCTRL|MODULE} specified
DMSWLK010E	Premature EOF on file *fn ft*
DMSWLK065E	{PRINT|MODULE} option specified twice
DMSWLK842E	No {control|library} file name found in *fn ft [fm]*
DMSWLK843I	An invalid control record was found and ignored:
DMSWLK844E	No linkedit performed
DMSWLK845W	Errors were encountered during the link edit processing that will probably make the loadlib unusable
DMSWLK846I	LKED *target_module* into *library*

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

VMFLOAD MODULE

Use the VMFLOAD command to generate a new CP nucleus, CMS nucleus, GCS nucleus, or stand-alone dump program. The VMFLOAD program uses a loadlist EXEC file, a control file, and an optional national language identifier to produce a punch file that has several object modules. VMFLOAD requires a virtual machine with at least 512K.

Command Format

VMFLOAD	*loadlist* *ctlfile* [*langid*]

loadlist
is the filename of an EXEC file that contains the names of object modules in the order in which they are to reside in the complete load file for the nucleus. The entries in the loadlist determine which files VMFLOAD punches and how VMFLOAD punches them (with or without a header card). The object modules are punched in the order specified, beginning at the top of the loadlist EXEC. A sample entry in a loadlist may look like:

```
&CONTROL OFF
&1 &2 &3 fn [ft] [(LANG]
 .
 .
 .
```

In each loadlist entry:

fn
is the filename of an object module to be punched.

ft
is the filetype of an object module to be punched. If *ft* is specified, the module is punched without a header card. If *ft* is not specified, the module is punched with a header card.

(LANG
is a special option that you use for national language-related files (eg. message respositories). Any entry with the LANG option is punched with a header card, even if *ft* is specified. If you specify a langid on the VMFLOAD command, VMFLOAD does special processing with these entries to determine the actual filetype of the object module file to be punched.

ctlfile
is the filename of the control file. This is usually the same control file used to apply updates to modules using the VMFASM or UPDATE commands. This file identifies the highest level object module available, if VMFLOAD cannot determine a specific filetype from the

loadlist. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

langid
is the identifier for national language-related files that get loaded into the nucleus.

If you specify a langid on the command line, VMFLOAD does special processing to determine the filetype of any module in the loadlist that contains the LANG option. VMFLOAD constructs the filetype identifier TXT*langid* and **always** uses TXT*langid* as the filetype for any modules marked with LANG in the loadlist. (This is true even in cases where an entry is in the form **&1 &2 &3 fn ft (LANG)**.

For example, the CMS loadlist (CMSLOAD EXEC) contains an entry for the CMS message repository:

```
&CONTROL OFF
.
.
.
&1 &2 &3 DMSMES  TXTAMENG (LANG
.
.
.
```

Assume FRANC is the langid for the French translation of national language-related files. To build a CMS nucleus that contains the French CMS message repository, you can issue the following command:

vmfload cmsload dmssp franc

When VMFLOAD reads CMSLOAD and processes the line

```
&1 &2 &3 DMSMES TXTAMENG (LANG
```

it searches for the file DMSMES TXTFRANC.

How VMFLOAD Works

1. Before you invoke VMFLOAD, you may want to issue one of the following commands:

   spool pun to *
   spool pun to *userid*

   to transfer the punched output as a reader file to your own virtual machine or to another virtual machine. If you want to do any additional controls, you should write an exec procedure to do the control and invoke VMFLOAD from that exec.

   Before the files specified in the loadlist are punched, VMFLOAD issues the command:

```
SPOOL PUN CONT
```

to ensure that the punched files appear as one deck.

2. VMFLOAD processes entries in the loadlist from top to bottom. If VMFLOAD has to search the control file to determine the filetype of the object module, the filetypes are based on the update level identifiers in the control file. (These are the identifiers used by the VMFASM to assign filetypes to object decks).

   If you do not specify a langid on the VMFLOAD command, VMFLOAD searches the control file to determine the filetype for all loadlist entries having either of the following forms:

   ```
   &1 &2 &3 fn
   &1 &2 &3 fn (LANG
   ```

   Remember that updates applied to source files are applied from the bottom of the file towards the top. Therefore, VMFLOAD searches the control file from the top towards the bottom to locate the most recent update level.

   For example, if a control file contains the following records:

   ```
   TEXT MACS DMKMAC
   LOCAL FIX1
   SPEC  AUX1111
   PTF   C12567DK
   IBM1  AUXSP
   ```

   then, for each entry in the loadlist, the VMFLOAD search order is:

   fn TXTLOCAL
   fn TXTSPEC
   fn TXTIBM1.

   As soon as VMFLOAD locates a file, it punches it, then continues processing the next entry in the loadlist. If none of the above filetypes exist for the loadlist entry, VMFLOAD searches for filename TEXT. If there is no TEXT file, VMFLOAD displays a message and continues processing with the next entry in the loadlist.

   Note: When determining the filetypes of object modules to punch, VMFLOAD ignores records that have an update level identifier of PTF and searches for the next lowest level identifier.

3. When all the object modules are punched, VMFLOAD issues the commands:

   ```
   SPOOL PUNCH NOCONT
   CLOSE PUNCH
   ```

   If you spooled the punch to yourself before invoking VMFLOAD, the completed load deck is placed in your virtual card reader.

VMFLOAD

Input and Output Files

Disk Input Files

loadlist EXEC	Contains the filenames, and optionally filetypes or LANG options, of the object modules to be punched.
DMKLD00E LOADER	The loader, which should be the first entry in the loadlist EXEC.
object modules	With filetypes of TEXT or TXT*xxxxx*, where *xxxxx* is the update level identifier in a control file used by VMFASM to name the object module.

Punch Output File

load deck	Punched to your virtual machine.

Usage Notes

1. SPGEN EXEC uses VMFLOAD to generate a new CP, CMS, or GCS nucleus.

2. The distributed system uses the following loadlists:

Loadlist		Usage
APLOAD	EXEC	CP nucleus without V=R for an AP or MP system
AVLOAD	EXEC	CP nucleus with V=R for an AP or MP system
CPLOAD	EXEC	CP nucleus without V=R for UP system
VRLOAD	EXEC	CP nucleus with V=R for UP system
CPLOADSM	EXEC	CP nucleus with Small CP (no V=R) for UP system
CMSLOAD	EXEC	CMS nucleus
GCSLOAD	EXEC	GCS nucleus

3. The distributed system uses the following control files:

Control File		Usage
DMKSP	CNTRL	CP nucleus without FRET TRAP for UP system
DMKSPA	CNTRL	CP nucleus without FRET TRAP for AP system
DMKSPM	CNTRL	CP nucleus without FRET TRAP for MP system
DMKSPT	CNTRL	CP nucleus with FRET TRAP for UP system
DMKSPAT	CNTRL	CP nucleus with FRET TRAP for AP system
DMKSPMT	CNTRL	CP nucleus with FRET TRAP for MP system
DMSSP	CNTRL	CMS nucleus
CSISP	CNTRL	GCS nucleus

4. The default langid is AMENG (American English). The other defined langids are:

Langid	Language
KANJI	Kanji (Japan)
UCENG	Uppercase English
PORTG	Brazilian Portuguese
FRANC	French
GER	German

5. After you have punched a new nucleus with VMFLOAD, you can use the MOVEFILE command to move the nucleus to tape, or, if the nucleus is in your virtual card reader, you can IPL it by issuing:

   ipl 00c

   When you IPL the virtual card reader, the loader is read first, and it loads the rest of the object modules. If the loader is successful, the nucleus is written on disk, and the load map is spooled to the virtual printer. If you want to preserve a disk copy of the load map, you should spool your printer to your virtual card reader, then read the file onto disk.

6. To locate file(s), VMFLOAD searches all of your accessed disks using the standard search order, A through Z.

Messages

SYSTEM LOAD DECK COMPLETE
This message is displayed when all the files in the loadlist have been punched.

INSUFFICIENT OR INVALID ARGUMENTS
The command line was incorrectly entered.

NO CONTROL FILE
The control file could not be located.

ERROR IN CONTROL FILE
The control file contains an invalid record.

NO LOAD LIST
The loadlist could not be located.

ERROR IN LOAD LIST
The loadlist contains an invalid record.

fn ft NOT FOUND
No text file was found.

ERROR ON PUNCH
An error occurred punching a file.

CP Loadlist Requirements

The CP loadlist EXEC contains a list of CP modules that is used by the VMFLOAD procedures to punch the text decks for the CP system. All modules following DMKCPE in the list are pageable CP modules. Each 4K page in this area may contain one or more modules. Pageable modules must not span the 4K page boundaries. The module grouping is governed by SPB (Set Page Boundary) cards. An SPB card is a loader control card that forces the loader to start this module at the next higher 4K boundary. If more than one module is to be contained in a 4K page, only the first is preceded by an SPB card.

The loader inserts SPB cards automatically where they are needed; you need not insert SPB cards.

The position of two modules in the loadlist is critical. All modules following DMKCPE must be reenterable and must not contain any address constants referring to anything in the pageable CP area. DMKCKN must be the last module in the loadlist.

The last entry in a CP loadlist must be:

```
&1  &2  &3  LDT  DMKSAVNC
```

X'02'	LDT | not used | DMKSAVNC | blank

For more information, see the Loader Terminate Statement section in the *VM/SP CMS Command Reference.*

Note: For virtual=real (V=R) users only, the DMKSLC text file is required. Use the VRSIZE command (see "VRSIZE MODULE" on page 15-94) to generate DMKSLC before you issue the VMFLOAD command. (See the flowchart, Figure 12-1 on page 12-2 "Deciding Which Updating Procedures to Use.") The **VM/SP Planning Guide and Reference** *provides a formula for calculating the size of the V=R area.*

VMFMAC EXEC

The VMFMAC EXEC procedure updates macro libraries (MACLIBs). If you specify a control file, VMFMAC invokes the CMS UPDATE command to update the COPY and/or MACRO files listed in the control file, and then builds a new macro library from the resulting new versions of those files.

Command Format

VMFMAC	*libname* [*ctlfile*]

libname
is the filename of the macro library to be updated; the filetype must be MACLIB. *libname* is also the filename of an EXEC file that lists the names of the library members. The entries in *libname* EXEC must be in the following format:

```
&1 &2 fn1
&1 &2 fn2
.
.
.
```

where *fn1*, *fn2*, and so on, are the filenames of the MACRO and COPY files to be updated and included in the macro library.

ctlfile
is the filename of an optional control file (if one exists) that is used to apply the updates. The filetype must be CNTRL. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

How VMFMAC Works

The VMFMAC procedure works as follows:

1. VMFMAC locates *libname* EXEC and the control file, if you specified one. It also erases any existing files named NEWMAC MACLIB and NEWMAC COPY. Then VMFMAC begins reading the MACRO and COPY filenames from the EXEC file, beginning at the bottom.

2. If you specify a control file, then for each entry in *libname* EXEC, VMFMAC:

   a. Invokes the UPDATE command with the CTL and OUTMODE A1 options to apply the updates specified in the control file.

      - UPDATE stacks information from the control file in the console stack and prints the update log file.

- The OUTMODE A1 option specifies that the files created by the UPDATE command are written onto the A-disk with a filemode of A1.

b. Adds the updated MACRO or COPY file (\$*fn* MACRO or \$*fn* COPY) to the macro library NEWMAC MACLIB.

c. Adds the *fn* UPDATES file created by the UPDATE command to the file NEWMAC COPY.

d. Erases \$*fn* MACRO or \$*fn* COPY and erases *fn* UPDATES.

3. If there are no update files for a MACRO or COPY file specified in *libname* EXEC, the MACRO or COPY file is added to NEWMAC MACLIB in its current form. If you specify a control file, NEWMAC COPY is added to NEWMAC MACLIB. This file contains a history of the updates applied by VMFMAC.

4. If no errors occur during the procedure, then when all the macros have been added to NEWMAC MACLIB, NEWMAC MACLIB is renamed *libname* MACLIB. The current *libname* MACLIB, if it exists, is erased.

   If errors occur during the VMFMAC EXEC procedure (for example, if a MACRO or COPY file is not found) *libname* MACLIB is not erased, and the updated macro library retains the name NEWMAC MACLIB.

Input and Output Files

Disk Input Files

libname EXEC	Contains a list of MACRO and COPY files to be updated and/or included in *libname* MACLIB.
ctlfile CNTRL	Optional control file used by the UPDATE command.

Input may also include MACRO and COPY files to be updated and/or included in the macro library, plus miscellaneous auxiliary control files (*fn* AUX*xxxxx*) and update files.

Disk Output Files

libname MACLIB	The updated macro library.
libname COPY	Contains the UPDATES files produced by UPDATE command processing. Note that this output file is created only if you specify a control file on VMFMAC.

Printer Output Files: The printer is spooled with the CONT option. When VMFMAC completes, the printer file contains a copy of each MACRO or COPY file in the macro library.

If you specify a control file, the printer file also contains:

- A copy of the control file.
- For each updated MACRO or COPY file, the update log file produced by the UPDATE command.
- The *libname* COPY file, which contains the accumulated UPDATES files created by the UPDATE command.

Usage Notes

1. When a file with a MACRO filetype is added to a MACLIB, the *membername* is taken from the macro prototype statement. When a file with a COPY filetype is added to a MACLIB, the *membername* is taken from the filename of the COPY file. This will be $*fn* if updates were found; otherwise it is just *fn*. If you include a *COPY statement as the first record in the file, in the format:

   ```
   *COPY membername
   ```

   the MACLIB directory uses *membername* to name the COPY file.

2. If errors occur during VMFMAC processing, consult the NEWMAC COPY file printed by VMFMAC. If you can correct the errors involving one or two MACRO or COPY files, add these members to NEWMAC MACLIB using the MACLIB command. Then enter the following commands:

   erase *libname* **maclib**
   rename newmac maclib *fm libname* = =

   The current *libname* MACLIB is erased and NEWMAC MACLIB is renamed *libname* MACLIB.

Responses

fn { **COPY** / **MACRO** } **ADDED.**

The specified MACRO or COPY file has been added to the macro library.

libname COPY ADDED.

The *libname* COPY file, containing the update history of the MACLIB, has been added.

***** TYPE 'VMFMAC LIBNAME <CTL>' *****

The VMFMAC command was entered with no operand(s).

***** libname EXEC NOT FOUND *****

VMFMAC could not locate the EXEC file associated with the macro library.

***** ctlfile CNTRL NOT FOUND *****

VMFMAC could not locate the specified control file.

***** fn { COPY / MACRO } NOT FOUND *****

The *fn* member named in *libname* EXEC could not be located.

***** ERRORS UPDATING fn { COPY / MACRO } *****

fn { COPY / MACRO } NOT INCLUDED IN MACLIB

An UPDATE command error occurred for library member *fn*, and the file was not written into the MACLIB.

DUE TO PREVIOUS ERRORS, THE RESULT OF THIS MACLIB BUILD
IS CALLED 'NEWMAC MACLIB', libname MACLIB HAS
NOT BEEN REPLACED

One or more errors were encountered, and you must correct them and create the MACLIB yourself.

Messages

The following messages are issued by the UPDATE command:

DMSUPD178I (Applying|Updating) *fn ft fm*
DMSUPD181E No update files were found

VMFMERGE EXEC

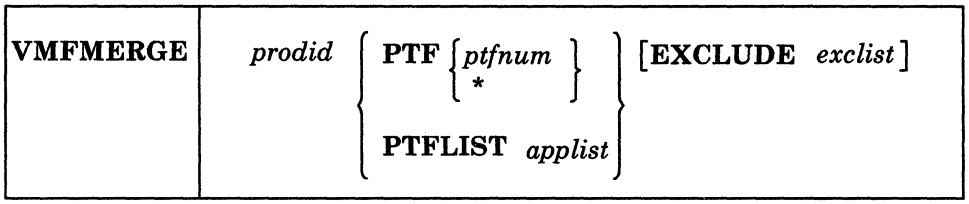
The VMFMERGE EXEC procedure applies PTFs (program temporary fixes) from the delta disk to the merge disk.

Do not use this procedure to service any of the base components of VM/SP. Use this procedure only when applying PTFs to Systems Network Architecture (SNA) products.

VMFMERGE requires a service control file (*ptfnum* SCF)[1] for each requested PTF and its requisites. The service control file contains instructions for applying the PTF.

To use VMFMERGE, you must access the minidisk containing the product parameter file, *prodid* VMFPARM. This file identifies the minidisks that VMFMERGE must access to service each product.

Command Format



prodid
is the id of the product. You can not specify EXCLUDE as the prodid, because EXCLUDE is a keyword for this EXEC.

PTF { *ptfnum* | * }
applies a single PTF if you specify a PTF filename (*ptfnum*). If you enter * instead of a PTF filename, this operand applies all of the PTFs for the product, as listed in the Apply List file named *prodid* APPLIST that is supplied on the service tape. You can not specify EXCLUDE as the ptfnum, because EXCLUDE is a keyword for this EXEC.

If the service tape contains an Exclude List file named *prodid* EXCLIST, any PTF listed in that file is not applied.

PTFLIST *applist*
applies the selected PTFs listed in the Apply List file named *applist* APPLIST. If you specify EXCLUDE as the Apply List filename, you can not use the EXCLUDE option to specify an Exclude List.

If the service tape contains an Exclude List file named *prodid* EXCLIST, any PTF listed in that file is not applied.

---

1 See "Service Control File" on page N-2 for a description.

EXCLUDE *exclist*
excludes the selected PTFs listed in the Exclude List file named *exclist* EXCLIST in addition to the PTFs listed in the Exclude List file named *prodid* EXCLIST, if that file is supplied on the service tape.

If you use *prodid* as the filename of your Exclude List, then VMFMERGE ignores your *prodid* EXCLIST during processing. Therefore, you should use a filename other than *prodid* for any Exclude List(s) that you create.

How VMFMERGE Works

To process PTFs, VMFMERGE:

1. Uses the product parameter file (*prodid* VMFPARM) to determine the virtual address of the merge and delta disks for the specified product.

2. Checks the Merge Log to make sure that the PTF you select is not already merged or superseded.

3. Reads the Service Control File to get the prerequisite and corequisite PTFs and the elements affected by this PTF.

   - If the Service Control File for any of the prerequisite or corequisite PTFs is missing, then processing for the current PTF stops.

   - If a prerequisite or corequisite PTF is in the Exclude List, then the current PTF is not merged.

   - If a prerequisite or corequisite PTF has a status of "SUPERSEDED," then that prerequisite or corequisite is not merged.

   - If the requisite is not within this product, the system displays a message indicating that the requisite PTF must be merged at some later time.

   - When you merge a PTF which is a requisite of a change in another product, be sure to note this requisite information. There is no automatic way to tell you of this cross product requisite, if at a later time you remove the change which is a requisite of a change in another product.

4. Does the necessary COPY/RENAME from the delta disk to the merge disk for each element in the prerequisite and corequisite chain that was not "SUPERSEDED" or already merged.

   Note: Temporary files are created during this COPY/RENAME process to insure system integrity. These files are erased during normal VMFMERGE processing.

5. Adds service history to the element if it is a text deck. Element history consists of text deck comment(s) containing:

   - The PTF or APAR number
   - A time and date stamp
   - Any apartext information that was in the Service Control File (SCF).

   Note: A temporary file ($APARTXT $VMFMERG) is created during this history process to ensure system integrity. VMFMERGE erases this file during normal processing.

6. Updates the Reqby Log to reflect all the requisite relationships of all the merged and superseded PTFs.

7. Marks in the Merge Log any ZAPs and PTFs that are superseded by this PTF. Those ZAPs and PTFs are never applied.

8. Puts entries in the Merge Log to show which PTFs have been merged.

Usage Notes

1. You can issue VMFMERGE to process a single PTF, a list of PTFs, or ALL PTFs on the input disk for a product.

2. Merged PTFs cannot be excluded, but can be superseded.

3. Superseded PTFs cannot be merged or excluded.

4. PTFs in the Exclude List can be superseded.

Messages

DMSWMG002E	File *fn ft [fm]* not found
DMSWMG008E	Device *vdev* invalid or nonexistent
DMSWMG017E	Invalid device address *vdev*
DMSWMG520E	Invalid operand: *operand*
DMSWMG545E	Missing operands
DMSWMG649E	Extraneous parameter *parm*
DMSWMG653E	Error executing *command*
DMSWMG856E	Disk address *vdev* is listed more than once on the DELTA and/or MERGE entry records in the *prodid* VMFPARM file.
DMSWMG857E	The number of disk addresses on the DELTA entry record cannot exceed nine.
DMSWMG858E	Unable to find a *tag* entry record in the *fn ft* file.
DMSWMG859E	The *prodid* VMFPARM file has no disk addresses on the {MERGE | DELTA} entry record.
DMSWMG860E	Only one {MERGE | DELTA} entry record may appear in the *prodid* VMFPARM file.
DMSWMG861I	Accessing *disk_type* disk *vdev* as *mode*.

DMSWMG862I	Change *name* has been *action*
DMSWMG863E	The MERGE disk *vdev* must be linked read-write.
DMSWMG864E	PTF *name* will not be *action* because it already is *status*
DMSWMG864I	PTF *name* will not be *action* because it already is *status*
DMSWMG864W	PTF *name* will not be *action* because it already is *status.*
DMSWMG865I	Processing PTF *name*
DMSWMG866W	No PTFs have been *action.*
DMSWMG867E	Invalid status *status* in *prodid* VMFMGLOG for entry *ptf.*
DMSWMG868E	PTF *name* is not a part of product *prodid.*
DMSWMG869E	Error in file *fn ft fm. data* is invalid for *tag* tag.
DMSWMG870E	Error in file *fn ft fm.* There are no elements.
DMSWMG871E	Error in file *fn ft fm.* The *name* tag is missing.
DMSWMG872E	Error in file *fn ft fm.* REPLACE tag missing after the element *name.*
DMSWMG873E	Error in file *fn ft fm. parm* is an invalid parameter. Expecting parameter(s) PRODID, PREREQ, COREQ, SUP, APARTEXT, or CHANGES
DMSWMG874E	Invalid entry found at line *line* in *fn ft*
DMSWMG882W	File *fn ft [fm]* [from *name*] not found on any DELTA disks from the VMFPARM file.
DMSWMG883W	PTF *name* is not a part of product *prodid* and must be *action* in product *prodid.*
DMSWMG886E	Filename *name* from the *fn ft [fm]* file is longer than 8 characters.
DMSWMG892E	PTF *name* has not been *action.*
DMSWMG893E	Incomplete processing, not all [required] PTFs were *action.*
DMSWMG893W	Incomplete processing, not all [required] PTFs were *action.*
DMSWMG898E	VMFREMOV processing is incomplete

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

VMFNLS EXEC

The VMFNLS EXEC procedure updates national language-related files. VMFNLS automatically applies updates to source files, generates text files, and renames them so they can be loaded into the system.

Command Format

VMFNLS	*fn ft ctlfile* [*(options ...[ )]*]

fn
is the filename of the source file that is to be converted to text.

ft
is the filetype of the source file that is to be converted to text. Only REPOS, DLCS, and ASSEMBLE are allowed.

ctlfile
is the filename of the control file that is used to apply updates to the source file before text is generated. The filetype must be CNTRL. The IBM-supplied control files are described in Appendix L, "Control File Identifiers" on page L-1.

options...
are options for three of the commands that VMFNLS can issue: GENMSG, CONVERT COMMANDS, and ASSEMBLE (via VMFASM). Some options are assigned by VMFNLS; other options that may be used depend on the filetype of the source file. See "How VMFNLS Works" below. (Refer to the *VM/SP CMS Command Reference* for complete descriptions of the commands and their options.)

How VMFNLS Works

The VMFNLS EXEC procedure depends on the type of input source file.

***If the input source file is a REPOS (message repository) file or a DLCS (definition language for command syntax) file*:**

1. VMFNLS issues the UPDATE command with the PRINT option (no other UPDATE options are accepted) to apply updates to the source file and send the update log to the printer. The updated file *$fn ft* is produced. (For example, when the CP message repository DMKMES REPOS is updated, the file $DMKMES REPOS is produced.)

   This updated *$fn ft* file is a temporary file that gets erased before VMFNLS completes. If necessary, VMFNLS changes the filename of this temporary *$fn ft* file to match the filename required for the text file; however, it does not use the filetype. (For example, VMFNLS renames

the updated TSAF message repository from $ATSCMRC REPOS to ATSUME $PORTG.)

If VMFNLS cannot properly apply the updates, the printer is closed and the exec terminates.

2. VMFNLS then determines the langid associated with the source file. If the source filename is only six characters, VMFNLS assigns the langid AMENG as a default; otherwise, it extracts the country code from the 7th and 8th characters of the source filename.

   If the source filename contains a country code, VMFNLS searches a file named VMFNLS LANGLIST to find the langid. This file contains a list of all valid country codes, along with the associated langids and language names. Here is an example of what the file might look like:

```
A  KANJI   Kanji
B  UCENG   Uppercase English
C  PORTG   Brazilian Portuguese
D  FRANC   French
E  GER     German
```

Figure 15-3. VMFNLS LANGLIST file

   VMFNLS uses this list to convert the source filename to the text filename. For example, if DMKMESD REPOS is the French version of the CP message repository; the text file that is produced for this file is DMKMES TXTFRANC.

   If the country code from the source filename is not contained in the VMFNLS LANGLIST file, then all temporary files are erased, the printer is closed, and VMFNLS terminates.

3. Next, VMFNLS compiles the source file with the appropriate command.

   - If the source file is a message repository (filetype REPOS), VMFNLS issues the GENMSG command to produce a text file from this input file.

     VMFNLS sets up the input parameters for GENMSG as follows:

     fn is the filename of the file to be compiled. GENMSG places this name in the ESD card when the text file is produced. The possible filenames are:

Component	fn
CP	DMKMES
CMS	DMSMES
GCS	CSIMES
TSAF	ATSUME

     ft is the filetype of the file to be compiled.

applid corresponds to the first three characters of the filename of the source file:

Component	applid
CP	DMK
CMS	DMS
GCS	CSI
TSAF	ATS

langid is determined from the country code contained in the filename of the source file. AMENG is the default if there is no country code in the filename.

options depend on the component associated with the source file: for the CP repository, VMFNLS uses the GENMSG options **CP MARGIN 63**; for other components, VMFNLS uses the GENMSG option **MARGIN 63**.

Note: For a Kanji CP repository, VMFNLS uses the GENMSG options **CP DBCS**; *for other Kanji repositories, VMFNLS uses the GENMSG option* **DBCS**.

Other GENMSG options may be specified with the VMFNLS command if they do not conflict with the GENMSG options assigned by VMFNLS.

GENMSG produces a text file that has the same filename as the input file, and a filetype of TXT*langid*. GENMSG also produces a listing file that has the same filename as the text file; however, VMFNLS changes the filename of the listing file to match the filename of the source file.

The following table shows some examples of the text and listing files written to the user's A-disk from a given message repository source file:

Source File	Text File	Listing File
DMKMES REPOS	DMKMES TXTAMENG	DMKMES LISTING
DMSMESA REPOS	DMSMES TXTKANJI	DMSMESA LISTING
CSIMESB REPOS	CSIMES TXTUCENG	CSIMESB LISTING
ATSCMRC REPOS	ATSUME TXTPORTG	ATSCMRC LISTING

The updated source file is then printed.

For a complete description of the GENMSG command, refer to the *VM/SP CMS Command Reference*.

- If the source file is a definition language for command syntax file (filetype DLCS), VMFNLS issues the CONVERT COMMANDS command to produce two text files from this input file.

  VMFNLS sets up the input parameters for CONVERT COMMANDS as follows:

  fn is the filename of the file to be compiled.

  ft is the filetype of the file to be compiled.

  fm is always set to *.

  options are not assigned by VMFNLS; any valid CONVERT COMMANDS options may be specified.

  CONVERT COMMANDS produces two text decks. The filenames of these text decks depend on the :DLCS statement contained in the input file. This statement identifies the applid, langid, and whether the input file is a user or system DLCS file.

  For a system DLCS file, the filenames of the text decks are *applid*SPA for the command syntax definition file and *applid*SSY for the translation and synonym table. For a user DLCS file, the filenames of the text decks are *applid*UPA for the command syntax definition file and *applid*USY for the translation and synonym table.

  CONVERT COMMANDS assigns the filetype TXT*langid* to the text files.

  The following table shows some examples of the text files produced from a given source file:

Source File	Text Files
DMSSPA DLCS	DMSSPA TXTAMENG DMSSSY TXTAMENG
DMSSPAB DLCS	DMSSPA TXTUCENG DMSSSY TXTUCENG
DMSSPAC DLSC	DMSSPA TXTPORTG DMSSSY TXTPORTG

  The updated source file is then printed.

  For a complete description of the CONVERT COMMANDS command, refer to the *VM/SP CMS Command Reference.*

4. Finally, VMFNLS modifies the text file to contain the summary of updates.

   The summary of updates is appended to the front of the text file that is produced. As a result, when the load file is loaded into storage by the loader, this update summary appears in the load map.

If the input source file is an ASSEMBLE file:

1. VMFNLS invokes the VMFASM EXEC to apply updates to the source, send the update log to the printer, and produce an associated text deck with a filetype of TEXT. The listing that results from the assembly is also printed.

   VMFNLS does not assign any ASSEMBLE options, but accepts the ASSEMBLE options that are VMFASM defaults (see page 15-38). Non-default ASSEMBLE options listed under VMFASM may be specified with the VMFNLS command.

2. VMFNLS determines the langid associated with the source file in the same way as described under item 2 on page 15-68.

   The following table shows some examples of the text files produced from a given source file:

Source File	Text File
DMSTRT ASSEMBLE	DMSTRT TXTAMENG
DMSTRTB ASSEMBLE	DMSTRT TXTUCENG
DMSTRTC ASSEMBLE	DMSTRT TXTPORTG

Messages

DMSWNL001E	No filename specified
DMSWNL023E	No filetype specified
DMSWNL032E	Invalid filetype *ft*
DMSWNL122E	Return code *rc* from *routine*
DMSWNL328E	Control file not specified
DMSWNL448E	Country code *code* not in *list*

For a complete explanation of each message, refer to *VM/SP System Messages and Codes.*

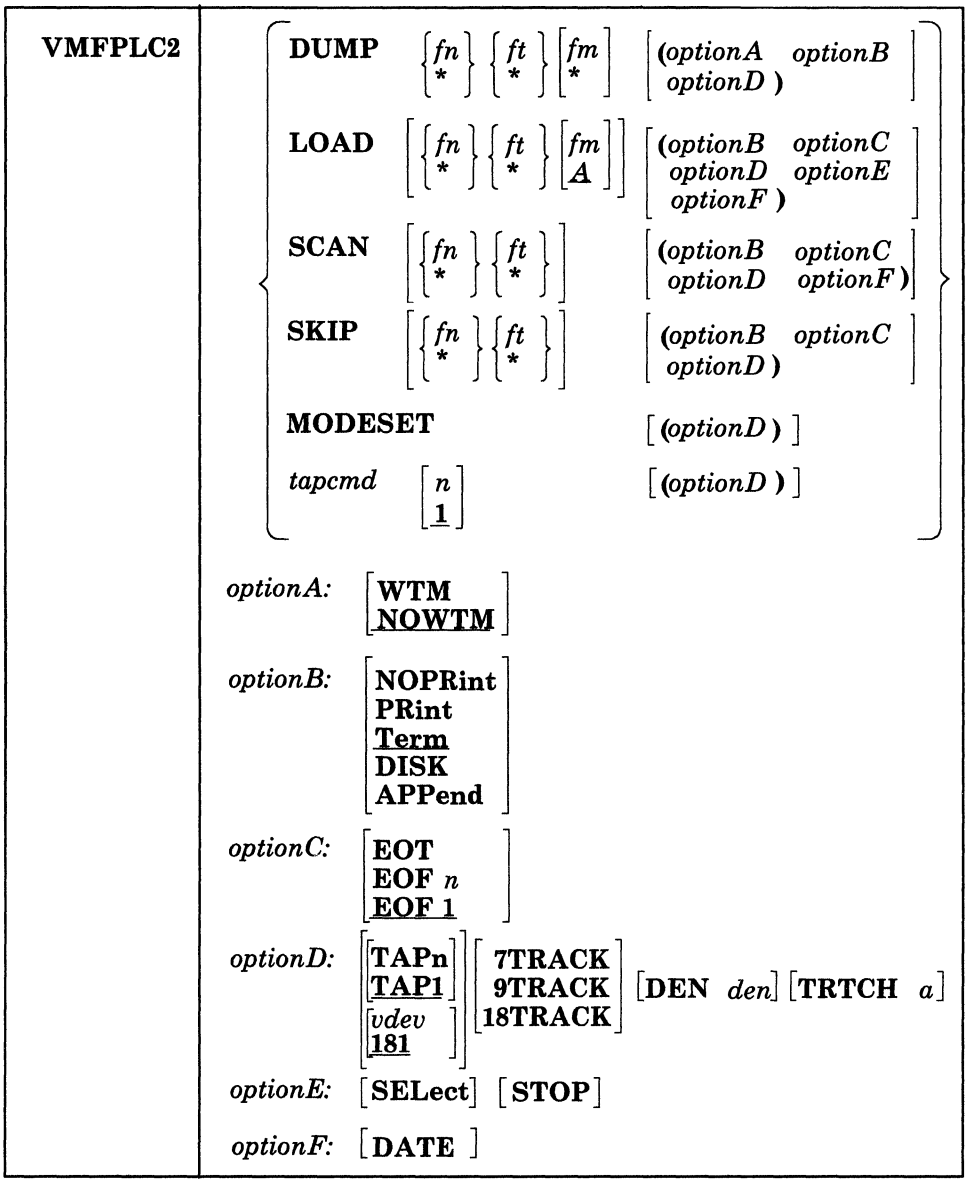
VMFPLC2 MODULE

The VMFPLC2 command:

- Loads source code from the Product Tape
- Loads the service installation VMSERV EXEC from the Program Update Tape
- Dumps CMS-formatted files from disk to tape. (Disk files to be dumped can contain either fixed-length or variable-length records.)
- Loads previously dumped files from tape to disk
- Does various control operations on a specified tape drive.

Note: The VMFPLC2 command does not process multivolume files. Files processed by the VMFPLC2 command must be CMS-formatted.

Command Format



DUMP $\left\{\begin{matrix} fn \\ * \end{matrix}\right\}\left\{\begin{matrix} ft \\ * \end{matrix}\right\}\left[\begin{matrix} fm \\ * \end{matrix}\right]$

dumps one or more disk files to tape. If *fn* and/or *ft* is specified as an asterisk (*) all files that satisfy the other file identifier(s) are dumped.

If *fm* is coded as a letter, that disk and its extensions are searched for the specified file(s). If *fm* is coded as a letter and number, only files with that mode number and letter (and the extensions of the disk referenced by that mode letter) are dumped. If *fm* is coded as an asterisk (*), all accessed disks are searched for the specified file(s). If *fm* is not specified, only the A-disk and its extensions are searched.

LOAD [{*fn* / *} {*ft* / *} [*fm* / A]]

reads tape files onto disk. If a file identifier is specified, only that one file is loaded. If the option EOF *n* is specified and no file identifier is entered, *n* tape files are written to disk. If *fn* or *ft* is coded as an asterisk (*), all files within EOF *n* that satisfy the other file identifier are loaded.

The files are written to the disk indicated by the filemode letter. The filemode number, if entered, indicates that only files with that filemode number are to be loaded.

SCAN [{*fn* / *} {*ft* / *}]

positions the tape at a specified point, and lists the identifiers of the files it scans. Scanning occurs over *n* tape marks, as specified by the option EOF *n* (the default is 1 tape file). However, if *fn* and *ft* are specified, scanning stops upon encountering that file, with the tape positioned to load the specified file.

SKIP [{*fn* / *} {*ft* / *}]

positions the tape at a specified point and lists the identifiers of the files it skips. Skipping occurs over *n* tape marks, as specified by the option EOF *n* (the default is 1 tape mark). However, if *fn* and *ft* are specified, skipping stops after encountering that file, with the tape positioned immediately following the file.

MODESET

sets the values specified by the DEN, TRACK, and TRTCH options, if the tape is at loadpoint. After initial specification in a VMFPLC2 command, these values remain in effect for the virtual tape device until they are changed in a subsequent VMFPLC2 command or in a RDTAPE, WRTAPE, or TAPECTL macro.

tapcmd [*n* / 1]

specifies a tape control function (*tapcmd*) to be executed *n* times. If *n* is not specified, the default is 1. These functions also work on tapes in a non-CMS format.

Tapcmd	Action
BSF	Backspace *n* tape marks
BSR	Backspace *n* tape records
ERG	Erase gap (defective section)
FSF	Forward-space *n* tape marks
FSR	Forward-space *n* tape records
REW	Rewind tape to load point
RUN	Rewind tape and unload
WTM	Write *n* tape marks

options...

If conflicting options are specified, the last one entered is in effect.

WTM
writes a tape mark on the tape after each file is dumped.

NOWTM
writes a tape mark after each file is dumped, then backspaces over the tape mark so that subsequent files written on the tape are not separated by tape marks.

NOPRINT
does not spool the list of files dumped, loaded, scanned, or skipped to the printer.

PRINT
spools the list of files dumped, loaded, scanned, or skipped to the printer.

TERM
displays a list of files dumped, loaded, scanned, or skipped at the terminal.

DISK
creates a disk file called TAPE MAP A5 containing the list of files dumped, loaded, scanned, or skipped.

APPEND
adds the TAPE MAP disk file (containing the list of files dumped, loaded, scanned, or skipped) to the end of an existing TAPE MAP file.

EOT
reads the tape until an end-of-tape indication (two tape marks) is recognized.

EOF *n*
reads the tape through a maximum of *n* tape marks. The default is EOF 1.

[**TAP***n*]
[*vdev*]
specifies the symbolic tape identification (TAP*n*) or the virtual device address (*vdev*) of the tape to be read from or written to, where *n* is 0-F. The default is TAP1 or 181. The unit specified by *vdev* must already be attached to your CMS virtual machine before any tape I/O operation can be attempted. The following symbolic names and virtual device addresses are supported.

Symbolic Name	Virtual Address	Symbolic Name	Virtual Address
TAP0	180	TAP8	288
TAP1	181	TAP9	289
TAP2	182	TAPA	28A
TAP3	183	TAPB	28B
TAP4	184	TAPC	28C
TAP5	185	TAPD	28D
TAP6	186	TAPE	28E
TAP7	187	TAPF	28F

7TRACK
specifies a 7-track tape. Odd parity, data convert on, and translate off are assumed unless TRTCH is specified.

9TRACK
specifies a 9-track tape.

18TRACK
specifies an 18-track tape.

DEN *den*
is the tape density, where *den* is 200, 556, 800, 1600, or 6250. If 200 or 556 is specified, 7TRACK is assumed. If 1600 or 6250 is specified, 9TRACK is assumed; if 800 is specified, 9TRACK is assumed unless 7TRACK is specified. If 38K is specified, 18TRACK is assumed. In the case of either 800/1600 or 1600/6250 dual-density drives, 1600 is the default if the 9TRACK option is specified. The tape drive operates at whatever tape density it was last set to if you do not specify either the 9TRACK option or the DEN option.

The following densities are allowed for the given track sizes.

7track 200, 556, 800

9track 800, 1600, 6250

18track 38K

TRTCH *a*
is the tape recording technique for 7-track tape. If TRTCH is specified, 7TRACK is assumed. One of the following must be specified as *a*:

a	Meaning
O	Odd parity, data convert off, translate off
OC	Odd parity, data convert on, translate off
OT	Odd parity, data convert off, translate on
E	Even parity, data convert off, translate off
ET	Even parity, data convert off, translate on

SELECT
inhibits loading of a file from the tape that causes replacement of an *identical* file on the disk. Files will be loaded only if they do not exist on the specified disk, or when the date/time stamp for the file on the disk *does not match* the date/time stamp for the corresponding file on the tape.

STOP
assumes that files contained on the tape are in alphabetical sequence. If the requested file is on the tape, the file is loaded onto disk and the tape stops. If the file is not on the tape and a file is encountered that is alphabetically beyond the bounds of the requested file, the tape stops. You must specify *fn ft*; neither *fn* nor *ft* may be specified as *.

DATE
displays listfile information during a SCAN. The information displayed includes number of records, length of records, and date/time stamp.

Usage Notes

1. Tape records written by the VMFPLC2 command are 4005 bytes long. These are not the same as CMS file records, which may be 512, 1K, 2K, or 4K bytes long. The first character is a binary 2 (X'02'), followed by the characters CMS and a file format byte, followed by 4000 bytes of file data packed without regard for logical record length. If a null block is dumped, the character "0" replaces the byte after CMS. This causes subsequent loading of null blocks to be ignored. In the final record, the character N replaces the blank after CMS, and the data area contains CMS file directory information.

2. If a tape file contains more CMS files than would fit on a disk, the tape load operation may terminate if there is not enough disk space to hold the files. To prevent this, when you dump the files, separate CMS (logical) files by tape marks, then forward space to the appropriate CMS file.

3. The CMS file directory is the first CMS record of each CMS file on tape.

4. It is possible to run a tape off. If you specify EOF *n*. FSF *n*, or FSR *n*, where *n* is greater than the number of tape marks or tape records on the tape, the tape will run off the reel.

5. If a BSF *n* or BSR *n* command causes the tape to return to loadpoint (for example, you enter: VMFPLC2 BSF 3, when there is only one tapemark to backspace over), then the message *DMSVMF110S ERROR READING 'TAPn {(cuu)}'* is issued.

6. The options for the 8809 tape drive must be 9TRACK and DEN 1600. Note that these are the default values, so you do not need to specify them.

7. When the STOP option is used with VMFPLC2 LOAD and the file does not exist, a return code of 44 is issued with message DMSVMF002E FILE 'fn ft fm' NOT FOUND.

8. Tapes with a density of 38k bpi and 18 tracks, are used only by the 3480 Magnetic Tape Subsystem. This subsystem does not read or write current half-inch tape (such as used by the 2400 and other 34xx subsystems.) Data on a 3480 cartridge must be copied onto a half-inch tape for interchanging with systems that do not have a 3480 subsystem.

9. VMFPLC2 is loaded as a nucleus extension.

10. For more information on tape file handling, see the *VM/SP CMS User's Guide*.

Messages

DMSVMF002E	FILE *fn ft fm* NOT FOUND
DMSVMF003E	INVALID OPTION *option*
DMSVMF010E	PREMATURE EOF ON FILE NUMBER *nn*
DMSVMF014E	INVALID FUNCTION *function*
DMSVMF017E	INVALID DEVICE ADDRESS *cuu*
DMSVMF023E	NO FILETYPE SPECIFIED
DMSVMF027E	INVALID DEVICE *device name*
DMSVMF029E	INVALID PARAMETER *parameter* IN THE OPTION *option* FIELD
DMSVMF037E	DISK *mode* IS READ/ONLY
DMSVMF042E	NO FILEID SPECIFIED
DMSVMF043E	*TAPn(cuu)* IS FILE PROTECTED
DMSVMF047E	NO FUNCTION SPECIFIED
DMSVMF048E	INVALID MODE *mode*
DMSVMF057E	INVALID RECORD FORMAT
DMSVMF058E	END-OF-FILE OR END-OF-TAPE
DMSVMF070E	INVALID PARAMETER *parameter*
DMSVMF104S	ERROR *nn* READING FILE *fn ft fm* FROM DISK
DMSVMF105S	ERROR *nn* WRITING FILE *fn ft fm* ON DISK
DMSVMF110S	ERROR READING *TAPn(cuu)*
DMSVMF111S	ERROR WRITING *TAPn(cuu)*
DMSVMF113S	TAPn(cuu) NOT ATTACHED
DMSVMF115S	CONVERSION FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	DUAL DENSITY FEATURE NOT SUPPORTED ON DEVICE *vdev*

DMSVMF115S	TRANSLATION FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	7-TRACK FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	9-TRACK FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	18-TRACK FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	800 BPI FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	1600 BPI FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF115S	6250 BPI FEATURE NOT SUPPORTED ON DEVICE *vdev*
DMSVMF613E	VMFPLC2 must be invoked as a nucleus extension
DMSVMF671E	Error loading file *fn ft fm*; rc=*nn* from RENAME
DMSVMF701I	NULL FILE

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

VMFREMOV EXEC

The VMFREMOV EXEC procedure removes PTFs (program temporary fixes) applied by the VMFMERGE EXEC procedure.

Do not use this procedure to service any of the base components of VM/SP. Use this procedure only when applying PTFs to Systems Network Architecture (SNA) products.

Command Format

VMFREMOV	*prodid*	[**PTF** { *ptfnum* | * }] [**PTFLIST** *remlist*] [**CONVERT** [*lastfilemode*]]

prodid
is the id of the product.

PTF { *ptfnum* | * }
removes the specified PTF (*ptfnum* is the filename of a PTF) or, if you enter * instead of a ptfnum, removes the PTFs listed in a Remove List file named *prodid* REMLIST. This file must be previously created and must be on a delta disk.

PTFLIST *remlist*
removes the selected PTFs listed in a Remove List file named *remlist* REMLIST. This file must be previously created and must be on a delta disk.

CONVERT
invokes the tool that creates the Reqby Log file, if one does not already exist, but does **not** remove any changes.

lastfilemode
specifies the filemode of the last delta disk. VMFREMOV assumes that the merge disk is always accessed as E and that all other delta disks are accessed as consecutive filemodes between F and *lastfilemode*. If you do not specify *lastfilemode*, then the filemodes for the merge and delta disks are determined from the information in the *prodid* VMFPARM file. Only use the lastfilemode parameter if you *know* you have the correct disks accessed as the proper modes.

How VMFREMOV Works

To remove PTFs applied by VMFMERGE, VMFREMOV:

1. Obtains data from the Merge Log and the Service Control File(s) to build the Reqby Log if one does not already exist. The Reqby Log contains a list of all dependent PTFs which must be removed if their requisite PTF is removed.

   If a Service Control File for any of the PTFs is missing, then processing continues. However, the Reqby Log will be incomplete if the missing Service Control File contains requisites. If any SCFs were missing, processing ends after VMFREMOV completes the build of the Reqby Log. No PTFs are removed.

2. Checks the Merge Log to insure that the PTF to be removed is currently merged.

3. Reads the Reqby Log for the list of dependent PTFs to remove.

4. Removes a PTF (i.e UV00007) which may supersede other PTFs (i.e. UV00005). Note that the other PTF (UV00005) is no longer superseded and its status is what it was prior to the merge of the primary PTF (UV00007) (i.e.merged, superseded, or no status).

   If the prior status of UV00005 is no status, then VMFREMOV removes its dependents.

5. Copies, from the delta disk to the merge disk, each element affected by the PTF being removed, if previous service for an element is merged. If the element has not been merged, VMFREMOV erases the element from the merge disk.

   The very first line of a copied text deck is always a comment line consisting of the PTF name and the date and time stamp. Any information on the **:apartext** entry is copied, but the first line is a comment.

   Note: A temporary file (with filetype of OVMFMGLG) is created during this procedure to insure system integrity. This file is erased during normal VMFREMOV processing.

6. Removes the element's entry from the Reqby Log. If an element has other elements that are dependent upon it, VMFREMOV also removes those dependent entries from the Reqby Log.

7. Updates the Merge Log with the current status of the PTFs. The *merged* entry is commented out. Another comment is added to the end of the Merge Log (with a time and date stamp) indicating that the PTF has been removed.

   If a PTF that supersedes another PTF is removed, then the SUPERSEDED entry is commented out.

VMFREMOV

Note: If you remove a change, VMFREMOV has no way of knowing if the change is a requisite of a change in another product. You should have made note of any cross-product requisite information during VMFMERGE processing (see page 15-64) to know which changes to remove manually.

Messages

DMSWRM002E	File *fn ft [fm]* not found
DMSWRM002W	File *fn ft [fm]* not found
DMSWRM008E	Device *vdev* invalid or nonexistent
DMSWRM017E	Invalid device address *vdev*
DMSWRM520E	Invalid operand: *operand*
DMSWRM545E	Missing operands
DMSWRM632E	I/O error in EXECIO; RC-*nn* from *command* command
DMSWRM649E	Extraneous parameter *parm*
DMSWRM653E	Error executing *command*
DMSWRM823E	PTF *name1* is listed as a dependent of PTF *name2*, but PTF *name1* is not merged
DMSWRM824W	*prodid* VMFREQBY may be incomplete due to a missing SCF
DMSWRM856E	Disk address *vdev* is listed more than once on the DELTA and/or MERGE entry records in the *prodid* VMFPARM file.
DMSWRM857E	The number of disk addresses on the DELTA entry record cannot exceed nine.
DMSWRM858E	Unable to find a *tag* entry record in the *fn ft* file.
DMSWRM859E	The *prodid* VMFPARM file has no disk addresses on the {MERGE | DELTA} entry record.
DMSWRM860E	Only one {MERGE | DELTA} entry record may appear in the *prodid* VMFPARM file.
DMSWRM861I	Accessing *disk_type* disk *vdev* as *mode*.
DMSWRM862I	Change *name* {has been *action* | is no longer SUPERSEDED by *name*}
DMSWRM863E	The MERGE disk *vdev* must be linked read-write.
DMSWRM864W	PTF *name* will not be *action* because it is not *status*.
DMSWRM865I	Processing PTF *name*
DMSWRM866W	No PTFs have been *action*.
DMSWRM867E	Invalid status *status* in *prodid* VMFMGLOG for entry *ptf*.
DMSWRM874E	Invalid entry found at line *line* in *fn ft*
DMSWRM879W	Change name *name* appears more than once in the *fn ft*.
DMSWRM882E	File *fn ft [fm]* [from *name*] not found on any DELTA disks from the disks from the VMFPARM file.
DMSWRM883W	PTF *name* is not a part of product *prodid* and must be *action* in product *prodid*.
DMSWRM888E	Error in *name* SCF. No entry for element *fn ft*.
DMSWRM892E	PTF *name* has not been *action*.
DMSWRM893W	Incomplete processing, not all [required] PTFs were *action*.

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

VMFTXT EXEC

The VMFTXT EXEC procedure rebuilds a named text library (TXTLIB) file using a member list contained in an EXEC file with the same filename.

To locate the member files, VMFTXT searches all of your accessed disks using the standard search order, A through Z. The member list may specify an optional filetype for specific members.

Command Format

VMFTXT	*libname* [*ctlfile*]

libname
is the filename of the text library (TXTLIB) file that you want to update; it is also the filename of the EXEC file that contains the names of the library members. The recommended format of the EXEC file is as follows:

```
&TRACE OFF
*Optional comments may be included
 [&1] [&2] [&3] fn [ft] [FILename [) ] ]
 [&1] [&2] [&3] fn [ft] [FILename [) ] ]
   .
   .
   .
```

In each entry, *fn* and *ft* are the filename and filetype of an object file that you want to add to the library. Specifying the filetype is optional:

- If you specify *ft*, VMFTXT looks for the specific file.
- If you do not specify *ft*, and you do not specify a control file on the command line, VMFTXT looks for a filetype of TEXT.
- If you do not specify *ft*, but you do specify a control file on the command line, VMFTXT searches for the specified member in the filetypes determined by the control file.

Each entry in the member list EXEC file may also specify an optional filename parameter (FILename) to be passed directly to the TXTLIB command. This parameter indicates that the member name is to be taken from the filename and not from the CSECT name within the file.

ctlfile
is the filename of an optional control file that VMFTXT uses to determine the filetypes of the object files added to the text library. The filetype of the control file must be CNTRL.

This file is usually the same control file that is used with the VMFASM or UPDATE commands to apply updates to modules. The IBM-supplied control files are described in Appendix L, "Control File

Identifiers" on page L-1. The control file identifies the filetype search order if you do not specify the filetype in the member list.

How VMFTXT Works

For each entry in the member list that does not specify a filetype:

- If you do not specify a control filename on the command line, the default object module filetype is TEXT.
- If you do specify a control filename on the command line:

  1. VMFTXT searches the control file to determine the filetype of the object module. The filetypes are based on the update level identifiers in the control file. (The VMFASM EXEC uses these identifiers to assign filetypes to object decks.) Remember that updates are applied to source files from the bottom of the control file towards the top. Therefore, VMFTXT searches the control file from the top towards the bottom to locate the most recent update level.

     For example, assume that the control file contains the following records:

     ```
     TEXT MACS DMSMAC
     LOCAL FIX1
     SPEC AUX1111
     PTF C12567DK
     IBM1 AUXSP
     ```

     Then, for each entry in the member list, the VMFTXT search order is:

     fn TXTLOCAL
     fn TXTSPEC
     fn TXTIBM1
     fn TEXT

     Note: When determining the filetypes of object files to add to the library, VMFTXT ignores records that have an update level identifier of PTF and searches for the next level identifier.

  2. When VMFTXT locates a file, it adds it to VMFTXT TXTLIB, then continues processing the next entry in the member list. If there is no text file, VMFTXT displays a message and continues processing with the next entry in the member list.

  3. When the procedure successfully adds all the members, VMFTXT issues the commands:

     ERASE *libname* **TXTLIB A**
     RENAME VMFTXT TXTLIB A *libname* **TXTLIB A**

Input and Output Files

Disk Input Files

libname TXTLIB — the text library file to be updated.

libname EXEC — contains the filenames, and optionally the filetypes, of the object files to be included in the library.

object modules — with filetypes specified in the member list, or with filetypes of TEXT, or with filetypes of TXT*xxxxx* (where *xxxxx* is the update level identifier obtained from a control file).

Disk Work Files: The following file identifiers are reserved for use by VMFTXT:

```
VMFTXT CMSUT1
VMFTXT CMSUT2
VMFTXT TEXT
VMFTXT TXTLIB
```

If you have any of these files on your A-disk when you issue VMFTXT, VMFTXT issues an error message and processing ends with a return code of 28.

Note: If you do not need the VMFTXT TEXT and VMFTXT TXTLIB files for problem diagnosis, erase them. Edit the VMFTXT CMSUT2 file and issue the command shown, then erase the file.

Warning: Do NOT erase the VMFTXT CMSUT1 file.

If you are adding a member to the library that has a filetype other than TEXT, and you have a TEXT file for that member on your A-disk, then VMFTXT temporarily renames the file to VMFTXT CMSUT1 A during processing. When processing is complete, VMFTXT renames the file back to its original name. However, if processing is interrupted when there is a non-zero return code from VMFTXT (eg. hx or ipl cms), check for the VMFTXT CMSUT1 file on your A-disk. If the file exists, rename it to 'membername TEXT A'. (You can find out what the membername should be by looking in the VMFTXT CMSUT2 A file).

Messages

DMSWTX001E	No filename specified
DMSWTX002E	File *fn ft fm* not found
DMSWTX006E	No read/write A disk accessed
DMSWTX024E	File *fn ft fm* already exists
DMSWTX026E	Invalid parameter *parameter* for *function* function
DMSWTX056E	File *fn ft [fm]* contains invalid record formats
DMSWTX062E	Invalid character *char* in fileid *fn ft*
DMSWTX179E	Missing or duplicate MACS card in control file *fn ft fm*
DMSWTX183E	Invalid control file control card
DMSWMS405E	Invalid or missing message number *msgno*
DMXWTX895I	Member *fn ft* added
DMSWTX896E	File {*fn ft fm* | *fn* TEXT or *fn* TXT} not found
DMSWTX897E	Due to previous errors, the result of this TXTLIB build is called VMFTXT TXTLIB; your *fn* TXTLIB has not been replaced

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

VMFZAP EXEC

The VMFZAP EXEC procedure applies ZAPs and maintains a record of ZAPs in the ZAP Log.

Do not use this procedure to service any of the base components of VM/SP. Use this procedure only when applying PTFs to Systems Network Architecture (SNA) products.

VMFZAP uses the base disk, the merge disk, and the ZAP disk as inputs and produces an updated ZAP disk as output.

Command Format

VMFZAP	*prodid*

prodid
is the id of the product to be ZAPped.

How VMFZAP Works

When you issue the VMFZAP command, VMFZAP:

1. Uses the product parameter file (*prodid* VMFPARM) to determine the virtual addresses of the ZAP, merge, and base disks.

2. Reads the ZAP Log file (*prodid* VMFZPLOG) and builds a list of TEXT file names with ZAPs applied to them.

3. Erases all TEXT files in this list from the ZAP disk. ZAPs will be applied to the first version of these TEXT files found on the other accessed disks.

4. Erases the ZAP Log.

5. Reads the Merge Log file (*prodid* VMFMGLOG) and builds a list of ZAPs that are currently SUPERSEDED.

6. Reads the ZAP List file (*prodid* ZAPLIST) for the names of all ZAPs that you want to apply.

7. Checks each ZAP name to see if it is superseded. If it is not superseded, then VMFZAP reads the control file for that ZAP (*zapname* ZAP).

   The control file for a ZAP may contain information for ZAPping more than one TEXT file. VMFZAP separates this information by TEXT file name and processes the ZAP of each TEXT file in the order listed in the control file.

8. Checks each TEXT file name to make sure that it resides on some disk other than the ZAP disk. It is an error if the TEXT file already resides on the ZAP disk.

   If the TEXT file exists on another disk, VMFZAP writes a temporary file called $$VMFZAP ZAP to the ZAP disk containing the ZAP control records for the current TEXT file.

9. Copies the TEXT file to the ZAP disk under a temporary name using the format VMF$T*n* TEXT, where *n* is a number determined by how many TEXT files are affected by the ZAP currently being processed. When this temporary file has been successfully ZAPped, it is renamed to its original name on the ZAP disk.

10. Invokes the ZAPTEXT EXEC and passes the $$VMFZAP name and the VMF$T*n* filename to be ZAPped.

    For more information about ZAPTEXT, see "ZAPTEXT EXEC" on page 15-108.

11. Updates the ZAP Log with the information about the TEXT file that was just ZAPped.

12. Restores your disk search hierarchy after all ZAPs in the ZAP List have been processed.

Usage Notes

1. To use VMFZAP, you MUST have an A-disk accessed read/write. This disk MUST NOT be the ZAP disk, merge disk, or base disk. That is, the virtual address of your A-disk MUST NOT appear on the ZAP, Merge, or Base records of your VMFPARM file.

2. VMFZAP accesses the necessary disks using filemode letters E-N. The merge disk is accessed ahead of the base disk. When VMFZAP processing stops, your search hierarchy is restored.

Messages

DMSWZP002E	File *fn ft [fm]* not found
DMSWZP008E	Device *vdev* invalid or nonexistent
DMSWZP017E	Invalid device address *vdev*
DMSWZP520E	Invalid operand : *operand*
DMSWZP545E	Missing operands
DMSWZP649E	Extraneous parameter *parm*
DMSWZP653E	Error executing *command*
DMSWZP856E	Disk address *vdev* is listed more than once on the {BASE, ZAP | DELTA} and/or MERGE entry records in the *prodid* VMFPARM file.
DMSWZP858E	Unable to find a *tag* entry record in the *fn ft* file.
DMSWMG859E	The *prodid* VMFPARM file has no disk addresses on the {BASE | MERGE | ZAP} entry record.

DMSWMG860E	Only one {BASE | MERGE | ZAP} entry record may appear in the *prodid* VMFPARM file.
DMSWZP861I	Accessing *disk_type* disk *vdev* as *mode*.
DMSWZP862I	ZAP *name* has been *action*
DMSWZP863E	The ZAP disk *vdev* must be linked read-write.
DMSWZP864W	ZAP *name* will not be *action* because it already is *status*
DMSWZP865I	Processing ZAP *name*
DMSWZP874E	Invalid entry found at line *line* in *fn ft*
DMSWZP875E	File *fn ft [fm]* not found on any disks from the VMFPARM file.
DMSWZP876E	The total number of disk addresses on the BASE and MERGE entry records cannot exceed nine.
DMSWZP877W	*fn* TEXT was previously zapped but was not found on the ZAP disk.
DMSWZP878E	*prodid* ZAPLIST does not contain any unsuperseded zap names. No zaps will be applied.
DMSWZP879W	ZAP name *name* appears more than once in the *fn ft* [It will only be applied once.]
DMSWZP880E	Error in ZAPTEXT while processing *fn* TEXT. Text files affected by *filename2* ZAP will not be saved on the ZAP disk.
DMSWZP881E	*fn* TEXT was found on the zap disk but was not zapped during the VMFZAP run. This file should not be on the ZAP disk.
DMSWZP885I	File *prodid* VMFZPLOG not found on the ZAP disk. No text files will be removed from the ZAP disk.
DMSWZP886E	Filename *name* from the *fn ft [fm]* file is longer than 8 characters.
DMSWZP887E	Record number *number* from the *fn ft [fm]* file is longer than 80 bytes.

For a complete explanation of each message, refer to *VM/SP System Messages and Codes*.

VMSERV EXEC

The VMSERV EXEC procedure applies IBM preventive service to your system. VMSERV is included on the system Program Update Tape (PUT). You invoke VMSERV to install service for any or all products on the PUT; VMSERV invokes each product-specific service exec to load the service files and build the nucleus or module. You must invoke VMSERV for each physical volume of a customized PUT. We recommend that you use VMSERV to apply service to the products in the order they appear on the PUT.

Command Format

VMSERV	[{ **RESTART** / **BUILD** } *pnum* [*comp*]] [**NORESP**]

RESTART
starts processing a PUT volume at a particular point, usually after a previous service application caused VMSERV not to complete processing. VMSERV restarts at the *pnum* you specify. If you want, VMSERV continues servicing the rest of the products on that volume. You can specify RESTART each time you want to selectively load service.

BUILD
invokes a particular service exec to build a nucleus (or module), or invokes whatever build procedures are associated with the specified *pnum*. This operand assumes that all the service has been loaded from the PUT.

pnum
is the program number (type/model) of a product whose service appears on the PUT (i.e. 5664167). *pnum* is only valid when you specify RESTART or BUILD.

comp
is the component that the service exec is to begin applying service for, or, if you specify BUILD, the component to build. *comp* may be CP, CMS, IPCS, GCS, or TSAF. If the product inventory file (VMPUT PRODUCTS) entry is found for this product, then *comp* is added to the end of the operands specified in the entry. If you specify RESTART, then VMSERV passes *comp* only to the specified product's service exec and does not pass *comp* to any subsequent service execs.

NORESP
means no response. When you use this operand, VMSERV does not prompt you for the service minidisk addresses, but assumes that your system is set up as described in the installation guides for the various products.

We recommend that you specify NORESP as part of the entry in the VMPUT PRODUCTS file for the product you want to service. This ensures that VMSERV will always recognize the operand.

How VMSERV Works

When you receive a PUT:

1. Log onto MAINT.
2. Mount and ready the PUT.
3. Access your 191 minidisk as C.
4. Use VMFPLC2 to load the contents of the first physical tape file (VMSERV EXEC, VMFSTK EXEC, PUT Document, and PUT HEADER) to your C disk.
5. Issue the VMSERV command.
   a. VMSERV loads the contents of the second physical tape file (all the Memo(s)-to-Users and all the product level identifiers) to your C disk.
   b. VMSERV determines, from the filetype of the product level identifiers, what products' service is on the tape and how many logical tape files are associated with each product's service. VMSERV uses this information to 'map' the PUT and create a SERVICE DISKMAP file. During service processing, VMSERV updates this file to reflect service status (service loaded or build done) for each component.
   c. VMSERV asks whether or not you want to print all the Memo(s)-to-Users, then exits.
6. Issue the VMSERV command again to actually load all the service files from tape to disk.
   a. VMSERV verifies that the SERVICE DISKMAP file is 'mapped' (verifies the service level of the PUT).
   b. VMSERV then calls the product-specific service exec with the parameters you specify (and the parameters in the VMPUT PRODUCTS file). See the VMSERV command format above for a description of the parameters. Refer to "VMPUT PRODUCTS Inventory File" on page 11-8 for more information about the VMPUT PRODUCTS file.

      For VM/SP, the service exec:

      1) Loads the product service from the PUT.

2) Issues the appropriate error messages for all EXEC error conditions.
3) Lets you restart the service application after an interruption.

Refer to the PUT Memo(s)-to-Users for more details about the service exec.

c. After all service is loaded, VMSERV exits.

7. Issue VMSERV again with the BUILD operand. VMSERV invokes the product-specific build exec to build a new nucleus (or module).

If service processing fails for the product being serviced, the service exec exits with the appropriate return code.

Return Codes

0 **Service complete.** All Load and Build functions are done.
4 **Service loaded.** You no longer need the PUT for service processing. You may need to do a *build* depending on the product's packaging.
8 **Service halted.** Some (or no) service was loaded from the tape. You need to issue VMSERV with the *restart* option to apply service to this product. the tape position is known and is correct.
12 **Unknown error.** An unexpected error happened. You need to issue VMSERV with the *restart* option to continue. The status of service and the position of the tape are unknown.
xx **Unknown.** This is any unexpected return code from a service exec.

VRSIZE MODULE

The VRSIZE command builds a text file for generating a virtual=real (V=R) area in your system.

Command Format

VRSIZE	

How VRSIZE Works

If you build your CP nucleus with your own build exec and want to have a V=R area in your system, you must create a DMKSLC text file.

The VRSIZE command builds a file called DMKSLC TEXT. DMKSLC must be listed in the loadlist that you use to generate CP. DMKSLC comes right after DMKPSA (page zero) in the CP nucleus. It moves the set location counter from the beginning of page one to the end of the V=R area. The virtual=real size must be specified in thousands (K) rather than millions (M), such as 8192K instead of 8M. You need to create this file only once, as long as you do not want to change your V=R size. However, several different DMKSLC text files may be saved for different purposes.

To create the DMKSLC text file, enter the VRSIZE command and reply to the prompts, as in the example that follows:

vrsize

```
VIRTUAL=REAL OPTION REQUIRED (YES,NO):
```

yes

```
STORAGE SIZE OF VIRT=REAL  (MINIMUM IS 32K):
```

5120k (Specify your own V=R size here; the size must be in K.)

```
05120K STORAGE SIZE FOR VIRTUAL=REAL
IS THE ABOVE ENTRY CORRECT (YES,NO)
```

yes

```
Ready;
```

listfile dmkslc (date

```
FILENAME FILETYPE FM FORMAT LRECL RECS BLOCKS DATE     TIME
DMKSLC   TEXT     A1 F      80    3    1      mm/dd/yy hh:mm:ss
```

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. Once 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.

The format of the VSEVSAM command is:

VSEVSAM	

Example Using VSEVSAM

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 in the MACLIB, they can be erased from the disk after the MACLIB is created.

To invoke VSEVSAM EXEC, enter:

```
vsevsam
```

Once invoked, 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.
           If a default of TAPE is to be used, press "ENTER."
```

For this example we will read the macros from disk. Enter:

```
disk
```

The system responds:

```
DMSWVV790R If the default library name of "VSEVSAM" is to be
           used, press "ENTER."  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."
           Press "ENTER" to continue, else enter
           "QUIT" or the name to be used for the library.
```

Because 'vsammacs' is the name we want to call the library, press:

ENTER

The system responds:

```
DMSWVV808R Macro library libname 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 our A-disk, press:

ENTER

The system responds:

```
DMSWVV793I Maclib generation completed.

DMSWVV792R Are the macros to be erased from disk?
           Reply (YES | NO).  Press "ENTER" for default
           of "YES."
```

We want to erase the macros from disk, 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. If a default of TAPE is to be used, press "ENTER."
DMSWVV789W	Invalid response.
DMSWVV790R	If the default library name of "VSEVSAM" is to be used, press "ENTER." Else, enter the name to be used for the library.
DMSWVV791I	The library name will be *libname*. Press "ENTER" to continue, else enter "QUIT" or the name to be used for the library.
DMSWVV792R	Are the macros to be erased from disk? Reply (YES | NO). Press "ENTER" 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 S-disk to A-disk.

ZAP MODULE

ZAP is a CMS command that changes or dumps MODULE, LOADLIB, or TXTLIB files. It may be used to change either fixed or variable length MODULE files. It is for use by system support personnel only.

Input control records control ZAP processing. They can be submitted either from the terminal or from a disk file. Using the VER and REP control records, you can verify and replace data or instructions in a control section (CSECT). Using the DUMP control record, you can dump all or part of a CSECT, an entire member of a LOADLIB or TXTLIB file, or an entire module of a MODULE file.

Command Format

ZAP	{**MODULE** / **LOADLIB** / **TXTLIB**} [*libname1 ... libname3*] [(*option ...*[**)**]] *Options:* [**TERM** / **INPUT** *filename*] [**PRINT** / **NOPRINT**]

{**MODULE** / **LOADLIB** / **TXTLIB**}

indicates the type of file that you want to change or dump.

libname

is the filename of the library containing the member you want to change or dump. You can specify one to three library names. This operand is valid only for LOADLIB and TXTLIB files.

options...

TERM [**PRINT** / **NOPRINT**]

indicates that input to the ZAP service program is submitted through the terminal. If you specify TERM, the prompting message **ENTER:** is issued, and you can then enter input control records up to 80 characters long. If you specify PRINT with TERM, all output prints on the printer, but only error messages display at the terminal. If you specify NOPRINT with TERM, nothing prints on the printer, and all output except control records displays at the terminal. See Figure 15-4 on page 15-99.

INPUT *filename* [PRINT / NOPRINT]

specifies that input is submitted from a disk file called *filename*. This file must have a filetype of ZAP, and must be a fixed 80-byte sequential file residing on any accessible device. If you specify PRINT with INPUT *filename*, all output produced by the ZAP service program prints on the printer. In addition, commands in error, control records in error, and error messages display at the terminal. If you specify NOPRINT with INPUT *filename*, nothing prints on the printer, and all output displays at the terminal. See Figure 15-4.

	PRINT	NOPRINT
INPUT	Everything prints on the printer. Commands in error, control records in error, and error messages display on the terminal.	Nothing prints on the printer. Everything displays on the terminal.
TERM	Everything prints on the Printer. Error messages display on the terminal.	Nothing prints on the printer. Everything except control records displays on the terminal.

Figure 15-4. Valid Options and Their Output

Input Control Records

There are eight types of ZAP control records:

- DUMP
- NAME
- BASE
- VER or VERIFY
- REP
- LOG
- COMMENT
- END.

The ZAP program can accept only 80 characters of data for each control record. ZAP control records are free-form and need not start in position one of the record. Separate all information by one or more blanks. All address fields including disp (displacement) fields in VER and REP control records must contain an even number of hexadecimal digits, to a maximum of six digits (X'OD', X'02C8', X'014318'). Data fields in VER and REP control records must also contain an even number of hexadecimal digits.

If you want, you can separate the data anywhere by commas (for example, 83256482 or 8325,6482). The commas have no effect on the operation.

Note: Do not use blank spaces as separators within data fields.

The program sets the NOGO switch on if it finds a control record in error. A file can not be changed if the NOGO switch is turned on. The next valid NAME record turns the NOGO switch off. This means that if the control record is the NAME record, all succeeding records are ignored until the next NAME, DUMP, or END record. For any other error, only REP control records that follow are ignored.

DUMP Control Record

The DUMP control record allows you to dump a portion or all of a specified control section, or the complete member or module. The format of the output of the dump is hexadecimal with an EBCDIC translation of the hexadecimal data.

The DUMP control record is optional and resets the NOGO switch off. The DUMP control record must not immediately precede a BASE, VER, or REP control record. A NAME control record must precede the BASE, VER, and REP control records (if any) that follow a DUMP control record.

Format:

```
DUMP  {membername}  {csectname  [startaddress  [endaddress]]}
      {modulename}  {ALL                                    }
```

membername
is the name of the member you want to dump, or the member that contains the CSECT(s) you want to dump. This member must be in one of the libraries you specify on the ZAP command.

For a CMS TXTLIB, the format of the dump control record requires that you specify both *membername* and *csectname*. Because the library directory does not contain member names, any word may be used to replace *membername*. The program searches for only the second name following the dump operand; therefore, the second name must be *csectname*.

modulename
is the name of the module you want to dump, or the module that contains the CSECT(s) you want to dump. If you specify a module that has no loader table, the program dumps the entire module.

csectname
is the name of the control section that you want to dump. If you do not specify *csectname*, the program dumps only the first CSECT. *csectname* is required for CMS TXTLIBs, but is optional for OS TXTLIBs, LOADLIBs, and MODULE files. (See the discussion of csectname in "NAME Control Record" on page 15-101). You must not specify *csectname* for a module created with the NOMAP option.

startaddress
is the location within the specified CSECT where you want the dump to begin. This must be two, four, or six hexadecimal digits. The start address is the displacement from the beginning of the CSECT. For example, to start dumping at address 08 in a CSECT that begins at location X'400', you specify start address X'08', not X'0408'.

endaddress
is the last address you want to dump. This must be two, four, or six hexadecimal digits. If you do not specify *endaddress*, the program dumps from *startaddress* to the end of the CSECT. Note that start and end addresses apply only when you specify *csectname*.

ALL
tells the program to dump all CSECTs within the member or module you specify. You can specify ALL for MODULE files, LOADLIBs, and OS TXTLIBs, but not for CMS TXTLIBs. To dump all the CSECTs in a member of a CMS TXTLIB, you must issue a separate DUMP control record for each CSECT.

Usage Notes

1. Displacements listed in the dump output for a module file are calculated from the beginning location of the module. Therefore, the addresses in the output may differ from the displacements within a CSECT.

2. If a DUMP control record references a TXTLIB file that contains ORG statements causing more than one occurrence of an address, data found at the first occurrence is displayed, but any subsequent redefinition of data for the same location is ignored.

NAME Control Record

The NAME control record specifies the member or module and CSECT that contain the data you want the ZAP operation to verify or replace. The NAME control record must precede the BASE, VER, and REP control records. If it does not, the program sets the NOGO switch on.

Format:

NAME { *membername* / *modulename* } [*csectname*]

membername
is the member that you want to search for the desired CSECT.

modulename
is the module that you want to search for the desired CSECT.

csectname
is the name of the control section you want to change.

Usage Notes

1. You must specify *csectname* if the CSECT you want to change is in a CMS TXTLIB (that is, a TXTLIB created by the TXTLIB command from CMS TEXT decks that do not have a NAME card following the END card). The directory of a CMS TXTLIB contains only CSECT names and no member names. Select a word to replace *membername* as the first entry following the NAME operand in the NAME statement for a CMS TXTLIB.

   Note: The word you specify for the membername for a CMS TXTLIB should be a meaningful name. The filename of the LOG control record is determined by the membername or modulename you specify in the NAME control record.

2. The CSECT name you specify in the NAME record is compared with CSECT names in the directory. If the CSECT(s) match and no member name match is found, the member selected is the one that contains the CSECT name.

3. *csectname* is optional if the CSECT you want to change is a LOADLIB or an OS TXTLIB (that is, a TXTLIB created by the TXTLIB command from CMS TEXT decks that have a NAME card after the END card). The dictionaries of the specified libraries are searched for the member name and the member is then searched for the CSECT name, if you specified one. If you do not specify *csectname* for a LOADLIB or an OS TXTLIB, the program uses the first control section.

4. *csectname* is optional for a MODULE file. The module named in the NAME control record is found and, if you specify *csectname*, the first record is read to determine the number of records in the module and the availability of a loader table, which the program can then search for *csectname*. If you do not specify *csectname*, the program uses the beginning location of the module. You can not specify *csectname* if the module was created with the NOMAP option.

BASE Control Record

The BASE control record adjusts displacement values for subsequent VER or REP control records for a CSECT whose starting address is not location zero in an assembly listing.

The BASE control record is optional. See the discussion under "VER or VERIFY Control Record" on page 15-103. If you specify the BASE control record, it must follow the NAME record, but it need not follow the NAME record immediately. For example, you could have the following sequence of control records: NAME, VER, REP, BASE, VER, REP.

Format:

```
BASE   address
```

address
is the starting address of the CSECT. It must be two, four, or six hexadecimal digits.

Usage Note: If you do not specify a csectname in the NAME control record, you cannot specify any BASE value other than 00.

Example: For a CSECT starting at location X'400', you specify BASE 0400 in the BASE control record. If a subsequent VER card requests verification of location X'0408', BASE 0400 is subtracted from X'0408', and the program verifies location X'08' in the CSECT. This example applies if you specify TXTLIB, LOADLIB, or MODULE and the module map is present.

However, if no module map is present for a MODULE file (that is, the module was generated with the NOMAP option), then all operations are done as if the BASE address is location X'0'. For example, if you specify a BASE of X'400' and the address you want to look at or change in X'408', then you must specify X'08' and not X'408' in the REP and VER control records. The address in this case is from the start of the module.

VER or VERIFY Control Record

The VER control record requests verification of instructions or data within a CSECT. If the verification fails, the program does not do a subsequent REP operation until it finds another NAME control record.

The VER control record is optional. More than one VER record can follow a single NAME record.

Format:

```
{VERIFY}
{VER   }    disp    data
```

disp
is the displacement from the start of the CSECT containing the data to be inspected, if you did not submit a BASE control record for this CSECT. *disp* can also be the actual location of the data to be inspected, if you did submit a BASE control record. *disp* must be two, four, or six hexadecimal digits. This displacement does not have to be aligned on a fullword boundary. If this displacement value is outside the limits of the CSECT specified by the preceding NAME control record, the VERIFY control record is rejected.

data
is the data against which the data in the CSECT is compared. This must be an even number of hexadecimal digits.

Usage Note: If the VER control statement references data in a TXTLIB file that is later redefined by ORG statements, only the first data definition is verified.

Example: If the location you want to verify is X'3CC', and the CSECT begins at location X'2B0', you can enter:

```
base 02B0
ver 03CC data
```

or you can omit the BASE control record, subtract the CSECT start address from the address of the data, and enter:

```
ver 011C data
```

REP Control Record

The REP control record changes instructions or data at the specified location within the CSECT that you specify in a preceding NAME control record. The data specified in the REP control record replaces the data at the CSECT location specified by the disp operand. This replacement is on a "one-for-one" basis; that is, one byte of data defined in the control record replaces one byte of data at the location that you specified. If the replacement fails, the program does not do additional REP operations until it finds another NAME control record.

The REP control record is optional. More than one REP record can follow a single NAME record.

Format:

```
REP   disp   data
```

disp
is the displacement from the start of the CSECT of the data you want to replace (if you did not submit a BASE control record for this CSECT). *disp* can also be the actual location of the data if you did submit a BASE control record. *disp* must be two, four, or six hexadecimal digits. This displacement need not address a fullword boundary. If this displacement value is outside the limits of the CSECT being changed, the program does not do the replacement operation.

data
is the data that is to replace the data in the CSECT. This must be an even number of hexadecimal digits.

Usage Notes

1. Although you do not have to verify a location before replacing data, you should do so to make sure that the data being changed is what you expect it to be.

2. If the REP control statement references data in a TXTLIB file that is later redefined by ORG statements, the replacement of data takes place at the first occurrence of the data address of the TXTLIB member.

Example: If the location you want to replace is X'3CC', and the CSECT begins at location X'2B0', you can enter:

base 02B0
rep 03CC *data*

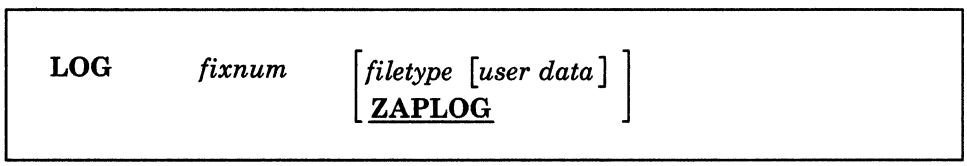
or you can omit the BASE control record, subtract the CSECT start address from the address of the data, and enter:

rep 011C *data*

LOG Control Record

The LOG control record lets you specify, after you apply a fix, a unique fix number which is recorded in a log file for the module or member. The file name of the log file is the same as *membername* or *modulename* in the NAME control record.

Format:



fixnum
specifies the number associated with the fix. Its length may vary from one-to-eight alphameric characters.

filetype
specifies the filetype of the log. The default is ZAPLOG.

user data
specifies any data that you want to enter into the log. If you specify *user data*, you must specify *filetype*.

Usage Notes

1. The LOG control record is optional and is allowed only if valid NAME and REP control records are found. The file name is obtained by the log routine from *modulename* or *membername* in the NAME control record. However, if no LOG control record is found, a dummy log record is written at the end of the user's valid REPs.

2. Log multiple names by including a LOG control record after each name. If the LOG record is not included after each name, error message DMSZAP070E results. Processing continues after the error messages occur.

3. The LOG record is 80 bytes in length and contains the following information:

   - Columns 1-63 contain the fixnum and, if specified, the filetype and user data.
   - Columns 64-80 contain the date and time of the ZAP.

COMMENT Control Record

The ZAP program ignores COMMENT control records. If the PRINT option is in effect, the program prints the comments.

Format:

```
* comment
```

There must be at least one blank following the asterisk (*) before you enter the text.

END Control Record

The END control record ends ZAP processing. The END record is required and must be the last control record for input from the console.

Format:

```
END
```

Special Considerations for Using the ZAP Service Program

Before using the ZAP command against MODULE files, you can use the MODMAP command to determine whether a module map exists and what it contains.

When a ZAP input file has more than one pair of VER and REP control records, and a VER control record (other than the first) fails, you must remove the records prior to the failing record and correct the error before you issue the ZAP command again. Otherwise, the file being changed returns to its original status.

The REP control record can not be used to place data in an undefined area such as a Define Storage area. If any part of a data field specified in a pair of VER and REP control records is an undefined area, the system displays warning message DMSZAP248W, and no data replacement occurs. If you do not issue a VER control record prior to the REP control record, some change to data may result. User-defined data may be inserted in undefined areas of text files by using the REP statement described under the LOAD command. (See the *VM/SP CMS Command Reference.*)

If the file to be dumped contains undefined areas (such as a DS or ORG statement in a TXTLIB member), the hexadecimal portion of the dump contains blanks to indicate that the corresponding positions are undefined.

VER and REP control words can be used to change TXTLIB members produced by FORTRAN compilers that store the length of the compiled text in the END card rather than in the ESD card. However, if a member of this type contains multiple CSECTs, only the first CSECT can be changed by the ZAP program.

The TXT records should be in ascending address order. If ZAP finds a TXT record with an address higher than the specified address, it stops scanning for the specified address. This means that ZAP control records affect only the data at the first occurrence of the address.

When applying ZAPs to a text deck created by a compiler, be aware that some compilers, such as FORTRAN, may generate a text deck in which the TXT records are not in ascending address order.

ZAPTEXT EXEC

The ZAPTEXT EXEC procedure modifies or dumps individual text files. Use ZAPTEXT like the ZAP service program, but only for text files, not MODULEs, TXTLIBs, or LOADLIBs, which ZAP processes. ZAPTEXT uses the same input control records as ZAP and also uses the EXPAND control record. The user's A-disk must be accessed as read/write to use ZAPTEXT.

Command Format

ZAPTEXT	*fn* [*ft* [*fm*]] [(*options:* ... [)]] *Options:* [**INPUT** *filename*] [**PRINT** / **NOPRINT**]

fn ft fm
is the fileid of the text file that you want to change. If you do not specify the filetype or filemode, the system assumes a filetype of TEXT and a filemode of A1. The filemode must specify a read/write disk.

options...

INPUT *filename*
specifies the name of a file that contains ZAP control records. This file must:

- Have a filetype of ZAP
- Be a fixed 80-byte sequential file that resides on any accessible disk.

If you do not specify a file, it defaults to whatever you specify as *fn* on the ZAPTEXT command.

PRINT
prints all output produced by ZAPTEXT on the printer. The system also displays error messages, commands in error, and control records in error.

NOPRINT
does not print anything on the printer, and displays all output at the terminal.

Input Control Records

ZAPTEXT uses the same input control records as the current ZAP service program, with the addition of the EXPAND control record. The ZAP service program ignores any EXPAND control records. Refer to "Input Control Records" on page 15-99 for information about the control records that both ZAP and ZAPTEXT can use. Use the ZAP control records with ZAPTEXT according to ZAP's TXTLIB conventions.

EXPAND Control Record

The EXPAND control record lets you increase the size of a named control section contained in the text file.

Format:

EXPAND	*csect size* [, *csect size* . . .]

csect
specifies the symbolic name of a control section whose length you want to increase.

size
specifies the decimal number of bytes for the system to add to the control section length. The system initializes the added bytes to binary zero. The maximum number of bytes for each control section that you indicate is 4095.

Usage Notes:

1. Each EXPAND control record may have multiple entries, but you must separate them with commas. Do not spill an entry onto the next line.

2. The system processes all EXPAND control records before any other control records, regardless of their position in the control file.

3. The effective length of the expansion, which is the actual number of bytes added to the control section, may be greater than the length that you specify for the expansion. This may occur if, after the specified expansion, the system must add padding bytes to align the next control section or common area.

4. When you increase a control section's size, it may affect the offset address of any following control section. This is important when you determine values for BASE, REP, and VER control records. Use the effective expansion lengths when you are determining control section offsets.

EXPAND Command

ZAPTEXT calls the EXPAND command if you specify an EXPAND control record in the ZAP control file. Use EXPAND to add space to a program in object deck form. The system creates object decks when you assemble or compile a source program. This is especially useful when you do not have the source code for a program or the program does not have a patch area.

Note: EXPAND can only add extra space at the end of named control sections (csects). EXPAND cannot expand private code (unnamed csect) and common areas (named or unnamed).

Do not increase the length of a program beyond its design limitations. For example, if you add space to a control section beyond the range of its base register addressability, that space is unusable.

Format:

EXPAND	*fn1* [*ft1* [*fm1* [*fn2* [*ft2* [*fm2*]]]]] [*(options: ... [ )]*] *Options:* [**INPUT** *filename* / **CSECT** *csect* **SIZE** *size*] [**PRINT** / **NOPRINT**]

fn1 ft1 fm1
identifies the input text file that the system expands. The file must have valid object deck information, like that created by an assembler or compiler. If you do not specify the filetype or filemode, the system assumes a filetype of TEXT and filemode of A1.

Note: EXPAND assumes that the input text file follows OS/VS standards and that the OS/VS Linkage Editor will accept it without error. The system does a limited check for errors. If the input file is invalid, the system may not expand the text file correctly.

fn2 ft2 fm2
identifies the output text file that the system creates. You can use an equal sign (=) for any of the file identifiers to indicate that it is the same identifier as *fn1*, *ft1* or *fm1*. The default fileid is $*fn1* = =. The system truncates the filename (*fn1*) to 7 characters before appending the $. In any case, *fm2* must be a read/write disk and can not be an asterisk (*).

options...

INPUT *filename*
specifies the name of an input file that contains EXPAND control records. If you do not specify INPUT, the filename defaults to the name of the text file that you are expanding (*fn1*). The filetype must be EXPAND. The system searches all accessed disks for this file.

Do not specify this option with the CSECT or SIZE options.

CSECT
specifies the symbolic name of a control section whose length the system will increase. If you specify CSECT, you must also specify SIZE.

Do not specify CSECT with the INPUT option.

SIZE
specifies the decimal number of bytes that the system adds to the control section length. The system initializes the added bytes to binary zeros. The maximum number of bytes for each control section is 4095. If you specify SIZE, you must also specify CSECT.

Do not specify SIZE with the INPUT option.

PRINT
prints on the printer all output that EXPAND produces. In addition, the system displays error messages, commands in error, and control records in error.

NOPRINT
does not print anything on the printer, and displays all output at the terminal.

System Response:

- After the system expands each CSECT that you specified, the system issues a message indicating the following:

  – The number of bytes added to the control section.

  The number of bytes added may be greater than the length that you specify for the expansion. This may occur if, after the specified expansion, the system must add padding bytes to align the next control section or common area.

  – The offset, relative to the start of the specified control section where the expansion began.

- If the system finds an error during processing, it stops the update and does not do the expansions.

Appendixes

Appendix A. Minidisks Reserved for the MAINT Userid

Figure A-1 lists the minidisks reserved for the MAINT userid. All minidisks except those marked with an asterisk (*) are defined in the sample system directory. Figure A-2 on page A-3 defines the default or recommended size of each minidisk according to DASD type.

DISK	VOLUME	USED BY	CONTENTS
190	VMSRES	CMS	System code (commands, EXECs, MACLIBs, TXTLIBs, and stand-alone service programs); updated TEXT decks (object modules)
191	VMSRES	CMS user	Installation EXECs and PROFILEs Work area
193	VMSRES	CMS IPCS GCS TSAF	System generation and maintenance EXECs; base code; updated TEXT decks Object code Interface files to IPCS Interface files to IPCS
194	VMSRES	CP	Object code
196	VMSRES	HPO	TEXT files
19D	VMPK01		HELP files
19E*			Extension disk for optional feature program products
201	VMSRES	EREP	Textlibs
293	VMSRES	CMS IPCS	PTFs; auxiliary and update files; updated TEXT; replacement CNTRL, EXEC, MODULE, and XEDIT files Replacement CNTRL, EXEC, MODULE, and XEDIT files
294	VMSRES	CP	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
295	VMSRES	CP/CMS user	System definition files Locally applied service
296	VMPK01	HPO	Service files
319	VMSRES		Licensed program "Memo to Users"
393	VMPK04	CMS	Source and MACRO files
394	VMPK04	CP	Source, MACRO, and COPY files
396	VMPK04	HPO	Source files

Figure A-1 (Part 1 of 2). Minidisks Reserved for the MAINT Userid

DISK	VOLUME	USED BY	CONTENTS
492	VMPK01	TSAF	Object code
493*	VMPK04	TSAF	Source code
494	VMPK01	TSAF	Replacement MACLIBs, TEXT, CNTRL, EXEC, and XEDIT files
496	VMPK01	IPCS	Service files
497	VMPK04	IPCS	Source code
595	VMPK01	GCS	Object code
596	VMPK01	GCS	Replacement MACLIBs, TEXT, CNTRL, EXEC, and XEDIT files
597*	VMPK04	GCS	Source code
59E	VMPK01	GCS	Extension disk for optional GCS applications

Figure A-1 (Part 2 of 2). Minidisks Reserved for the MAINT Userid

DASD Disk	3330	3340	3350	3375	3380	FBA
190	83	205	39	58	37	34312
191	22	54	10	15	10	8936
193	60	150	28	42	27	25080
194	60	150	28	42	27	25080
196	34	84	16	24	16	13952
19D[1]	110	275	50	77	48	45984
19E[2]						
201	50	123	23	35	23	20520
293	60	150	28	42	27	25080
294	47	115	22	33	21	19232
295	31	75	14	21	14	12544
296	41	100	19	28	19	16744
319	12	30	6	9	6	5000
393[3]	140	349	66	98	63	58520
394[3]	168	417	78	117	76	70000
396	74	183	35	52	34	30696
492	23	56	11	16	11	9400
493[3]	87	219	40	61	38	36000
494	23	56	11	16	11	9400
496	3	6	1	2	1	848
497[3]	14	35	7	10	7	5848
595	67	167	32	47	31	28000
596	47	115	22	33	21	19232
597[3]	83	206	38	58	36	34000
59E	22	54	10	15	10	9000

Figure A-2. Default or Recommended Minidisk Sizes[4]

[1]The allocation for this minidisk is larger than required to allow for future growth and for user-written and other selected product HELP files.

[2]The size of this minidisk depends on the requirements of the feature program products that are installed.

[3]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 your files, you may need to define this minidisk larger.

[4]The indicated sizes for some minidisks may not be adequate for the duration of the release cycle and may need to be increased at some time.

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.

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		

Figure B-1. Restricted Logon Passwords

Appendix C. Sample Installation/System Generation Profiles

Sample SPLOAD PROFILE

```
/*-------------------------------------------------------------------*/
/*                1600 BPI VM/SP Product Tape                        */
/*-------------------------------------------------------------------*/
/*                                                                   */
/* Group    Element     USERID   Address  Format   Volume  Tape File */
/*-------- --------    -------  -------- --------  ------  ----------*/
Install    Tools       MAINT      191    R5M05317   1         2
Sysgen     Tools       MAINT      193    R5M05317   1         3
System     Samples     MAINT      295    R5M05317   1         4
DASDMODL   Samples     MAINT      295    R5M05317   1         5
DASDTYPE   Samples     MAINT      295    R5M05317   1         6
CP         Object      MAINT      194    R5M05317   1         7
CMS        System      MAINT      190    R5M05317   1         8
CMS        Base        MAINT      193    R5M05317   1         9
/*                                                                */
IPCS       Object      MAINT      193    R5M05317   2         2
Help       Files       MAINT      19D    R5M05317   2         3
GCS        Interface   MAINT      193    R5M05317   2         4
GCS        Object      MAINT      595    R5M05317   2         5
TSAF       Interface   MAINT      193    R5M05317   2         6
TSAF       Object      MAINT      492    R5M05317   2         7
/*
CP         Source      MAINT      394    R5M05317   3         2
/*
IPCS       Source      MAINT      497    R5M05317   4         2
CMS        Source      MAINT      393    R5M05317   4         3
/*                                                                */
/*-------------------------------------------------------------------*/
/*                1600 BPI Source Code Tape                          */
/*-------------------------------------------------------------------*/
/*                                                                   */
GCS        Source      MAINT      597    R5M06036   1         2 *
TSAF       Source      MAINT      493    R5M06036   1         3 *
CMSFT      Source      MAINT      393    R5M06036   1         4 *
/*                                                                */
```

Figure C-1 (Part 1 of 2). Sample SPLOAD PROFILE

```
/*-------------------------------------------------------------------*/
/*        38K & 6250 BPI VM/SP Product Tape                          */
/*-------------------------------------------------------------------*/
/*                                                                   */
Install   Tools      MAINT     191   R5M05318   1        2
Sysgen    Tools      MAINT     193   R5M05318   1        3
System    Samples    MAINT     295   R5M05318   1        4
DASDMODL  Samples    MAINT     295   R5M05318   1        5
DASDTYPE  Samples    MAINT     295   R5M05318   1        6
CP        Object     MAINT     194   R5M05318   1        7
CMS       System     MAINT     190   R5M05318   1        8
CMS       Base       MAINT     193   R5M05318   1        9
IPCS      Object     MAINT     193   R5M05318   1       10
Help      Files      MAINT     19D   R5M05318   1       11
GCS       Interface  MAINT     193   R5M05318   1       12
GCS       Object     MAINT     595   R5M05318   1       13
TSAF      Interface  MAINT     193   R5M05318   1       14
TSAF      Object     MAINT     492   R5M05318   1       15
/*
IPCS      Source     MAINT     497   R5M05318   2        2 *
CP        Source     MAINT     394   R5M05318   2        3 *
CMS       Source     MAINT     393   R5M05318   2        4 *
/*                                                                   */
/*-------------------------------------------------------------------*/
/*              38K & 6250 BPI Source Code Tape                      */
/*-------------------------------------------------------------------*/
/*                                                                   */
GCS       Source     MAINT     597   R5M06037   1        2 *
TSAF      Source     MAINT     493   R5M06037   1        3 *
CMSFT     Source     MAINT     393   R5M06037   1        4 *
/*                                                                   */
/*-------------------------------------------------------------------*/
/*     38K & 6250 & 1600 BPI VM/SP NLS Feature Tape                  */
/*-------------------------------------------------------------------*/
/*                                                                   */
CMS       Base       MAINT     193   R5XXXXXX   1        2
CP        Object     MAINT     194   R5XXXXXX   1        3
Help      Files      MAINT     ?     R5XXXXXX   1        4
CMS       Source     MAINT     393   R5XXXXXX   1        5
TSAF      Object     MAINT     492   R5XXXXXX   1        6
GCS       Object     MAINT     595   R5XXXXXX   1        7
/*                                                                   */
/*                                                                   */
```

Figure C-1 (Part 2 of 2). Sample SPLOAD PROFILE

Sample SPGEN PROFILE

```
/*-------------------------------------------------------------*/
/*                 S P G E N   P R O F I L E                   */
/*                                                             */
/*               SYSTEM GENERATION PARAMETERS                  */
/*-------------------------------------------------------------*/
/*-------------------------------------------------------------*/
/*                        CP parameters                        */
/*-------------------------------------------------------------*/
:CP_small.      no           /* Small CP? (no, yes)          */
:CP_processor.  UP           /* UP, AP, MP?                  */
:CP_fret.       no           /* Fret trap? (no, yes)         */
:CP_vr.         no           /* V = R? (no, yes)             */
:CP_control.    *            /* Nucleus control file         */
:CP_loadlist.   *            /* Nucleus loadlist name        */
:CP_lang.       AMENG        /* Nucleus default language     */
:CP_mapname.    CPNUC MAP    /* Nucleus load map name        */
:CP_mapuserid.  MAINT        /* Load map destination         */
/*                                                           */
/*                                                           */
:CP_setup.
   ACCESS 295 N/A            /* Local modifications          */
/* ACCESS 294 O/A     */     /* Current PUT disk             */
   ACCESS 194 P/A            /* CP object code               */
/* ACCESS 394 Q/A     */     /* CP source code               */
   ACCESS 193 R/A            /* Sysgen tools                 */
/*-------------------------------------------------------------*/
/*                       CMS parameters                        */
/*-------------------------------------------------------------*/
:CMS_control.   DMSSP        /* Nucleus control file         */
:CMS_loadlist.  CMSLOAD      /* Nucleus loadlist name        */
:CMS_lang.      AMENG        /* Nucleus default language     */
:CMS_mapname.   CMSNUC MAP   /* Nucleus load map name        */
:CMS_mapuserid. MAINT        /* Load map destination         */
/*                                                           */
/*                                                           */
:CMS_setup.
   ACCESS 295 O/A            /* Local modifications          */
/* ACCESS 293 P/A     */     /* Current PUT disk             */
/* ACCESS 393 Q/A     */     /* CMS source code              */
   ACCESS 193 R/A            /* CMS object code              */
   ACCESS 190 N              /* CMS system disk              */
```

Figure C-2 (Part 1 of 2). Sample SPGEN PROFILE

```
/*-------------------------------------------------------------*/
/*                        GCS parameters                       */
/*-------------------------------------------------------------*/
:GCS_control.   CSISP        /* Nucleus control file          */
:GCS_loadlist.  GCSLOAD      /* Nucleus loadlist name         */
:GCS_lang.      AMENG        /* Nucleus default language      */
:GCS_mapname.   GCSNUC MAP   /* Nucleus load map name         */
:GCS_mapuserid. MAINT        /* Load map destination          */
/*                                                           */
/*                                                           */
:GCS_setup.
   ACCESS 595 Q/A            /* GCS object code               */
   ACCESS 193 R/A            /* Sysgen tools                  */
/*-------------------------------------------------------------*/
/*                      CPHPO parameters                       */
/*-------------------------------------------------------------*/
:CPHPO_control.  DMKH40      /* Nucleus control file          */
:CPHPO_loadlist. H40CPLOD    /* Nucleus loadlist name         */
:CPHPO_lang.     AMENG       /* Nucleus default language      */
:CPHPO_mapname.  CPHPONUC MAP/* Nucleus load map name         */
:CPHPO_mapuserid. MAINT      /* Load map destination          */
/*                                                           */
/*                                                           */
:CPHPO_setup.
   ACCESS 295 K/A            /* Local modifications           */
/* ACCESS 296 L/A     */     /* HPO PUT disk                  */
   ACCESS 196 M/A            /* HPO object code               */
/* ACCESS 294 N/A     */     /* CP PUT disk                   */
   ACCESS 194 O/A            /* CP object code                */
/* ACCESS 396 P/A     */     /* HPO source code               */
/* ACCESS 394 Q/A     */     /* CP source code                */
   ACCESS 193 R/A            /* Sysgen tools                  */
```

Figure C-2 (Part 2 of 2). Sample SPGEN PROFILE

Sample CMS Nucleus Generation Profile (DMSNGP ASSEMBLE)

```
SYS     TITLE   'DMSNGP - CMS Nucleus Generation Profile'
        PRINT   NOGEN
DMSNGP  CSECT
        DEFNUC  SYSDISK=190,          * S disk address                  X
                YDISK=19E,            * Y-disk address                  X
                HELP=19D,             * Help disk address               X
                LANGID=AMENG,         * Default is American English     X
                DBCS=NO,              * Default is not a DBCS lang      X
                LANGLEV=5,            * DCSS ID for multiple DCSS       X
                SAVESYS=YES,          * Using CMS in DCSS yes or no     X
                SYSNAME=CMS,          * Name of above DCSS to save      X
                USEINST=YES,          * Using EXEC/XEDIT in DCSS        X
                INSTSEG=CMSINST,      * Name of above DCSS to save      X
                REWRITE=YES,          * Write nucleus yes or no         X
                IPLADDR=190,          * Address of where to write       X
                CYLADDR=?,            * Cyl/Blk of where to write       X
                IPLCYL0=YES,          * write ipl text on cyl 0         X
                VERSION=,             * VM/SP REL n mm/dd/yy hh mm ss   X
                INSTID=               * VM/SP CONVERSATIONAL MONITOR SYS
        END
```

Figure C-3. Sample CMS Nucleus Generation Profile

Appendix D. Special Options for CP

This appendix contains information about the following options that you can select when generating a CP nucleus:

- Small CP
- CP FRET Trap
- Virtual=Real (V=R).

If you are using ITASK EXEC or SPGEN EXEC to build your CP nucleus, these options are selected by entries in SPGEN PROFILE (see "Sample SPGEN PROFILE" on page C-3).

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 may be required.

Small CP

Overview

The entry for this option in the sample SPGEN PROFILE is **no**.

The Small CP option allows you to reduce the size of the CP nucleus by removing specific types of support, such as virtual=real support, attached processor and multiprocessor support, and support for:

SUPPORT	MODULES
3066	DMKGRH
Remote 3270	DMKRGA, DMKRGB, DMKRGC, DMKRGD
3340 Alternate Track	DMKTRK
3375/3380	DMKDAD
3704/3705	DMKRNH
3800 Printers	DMKVSV
3850 MSS	DMKSSS, DMKSSU
MVS Guest	DMKFPS, DMKQVM, DMKVSC
SNA (CCS)	DMKVCP, DMKVCR, DMKVCT, DMKVCV, DMKVCX
TTY Terminal Support	DMKTTZ

Procedure

- Edit SPGEN PROFILE using the System Product Editor (XEDIT):
  - If you need **ANY** of the support listed above, you CAN NOT use the Small CP option. Make sure that the "CP_small" entry is **no**.
  - If you DO want to use the Small CP option, change the "CP_small" entry to **yes**.
  - Because Small CP does not support the Virtual=Real option, make sure that the "CP_vr" entry is **no**.
  - If you want to use the CP FRET Trap option, change the "CP_trap" entry to **yes**. See "CP FRET Trap" on page D-3 for more information.
- 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 (Chapter 3 or Chapter 4) that you have been using to do the installation.
  - If you are rebuilding the CP nucleus in an existing system, go to "Building the New CP Nucleus" on page K-1.

CP FRET Trap

Overview

The entry for this option in the sample SPGEN PROFILE is **no**.

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.

Control files and MACLIBs are provided for installing CP FRET Trap in uniprocessor (UP), attached processor (AP), and multiprocessor (MP) systems:

Control Files:

DMKSPT	CNTRL	*for UP systems*
DMKSPAT	CNTRL	*for AP systems*
DMKSPMT	CNTRL	*for MP systems*

MACLIBs:

DMKSPT	MACLIB	*for UP systems*
DMKSPAT	MACLIB	*for AP systems*
DMKSPMT	MACLIB	*for MP systems*

These MACLIBs contain an updated OPTIONS COPY file which has the option &FRETRAP changed from the default value of 1 to 0. SPGEN 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 CP FRET Trap code:

Text Files:

For UP systems:

DMKCPI	TXTFT
DMKFRE	TXTFT
DMKFRT	TXTFT

For AP and MP systems:

DMKCPI	TXTAPFT
DMKFRE	TXTAPFT
DMKFRT	TXTAPFT

For more information about using CP FRET Trap, refer to the *VM Diagnosis Guide.*

CP FRET Trap

Installation Procedure

Notes:

1. *This procedure uses SPGEN EXEC to generate the CP nucleus. If you are regenerating the CP nucleus in an existing system, and you want to install CP FRET Trap without using SPGEN, see the "Manual Installation Procedure" on page D-5.*

2. *If you want to remove CP FRET Trap, see the "Procedure to Remove CP FRET Trap" on page D-6.*

- Edit SPGEN PROFILE using the System Product Editor (XEDIT):
  - Change the "CP_fret" entry to **yes**.
  - If you also want to use the Small CP option or the Virtual=Real option, change the corresponding entry in SPGEN PROFILE ("CP_small" or "CP_vr") to **yes**. You CAN NOT generate a system using both Small CP and Virtual=Real, so make sure that the entry for the option you are not using is **no**. See "Small CP" on page D-2 and "Virtual=Real" on page D-10 for more information.
- 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 (Chapter 3 or Chapter 4) that you have been using to do the installation.
  - If you are rebuilding the CP nucleus in an existing system, go to "Building the New CP Nucleus" on page K-1.

Manual Installation Procedure

Note: This manual installation procedure is not valid if:

1. *You have made local modifications to DMKCPI, DMKFRE, or DMKFRT.*

   OR

2. *The control file that you normally use to build a new CP nucleus is a local control file and not one of the supplied control files (DMKSP, DMKSPA, or DMKSPM).*

Refer to "Considerations for Users with Local Modifications to CP" on page D-8.

- Make sure that you have a backup copy of your current system.
- In addition to the 295, 194, and 294 minidisks, you (MAINT) should have access to the CMS system disks. The search order is:

295	A	R/W	Local modifications minidisk
294	B/A	R/O	CP, PTFs, auxfiles, updates, updated text files, and MACLIBs from the latest PUT
194	C/A	R/O	CP text files and MACLIBs
190	S	R/O	CMS system minidisk
193	T	R/W	CMS base minidisk

- Build a new CP nucleus using the loadlist and CP FRET Trap installation control file that matches your system configuration:

  1. Make sure that your virtual card reader and punch do not contain any files:

     a. Close any active files in these devices by issuing:

        close punch
        close reader

     b. Empty your reader and punch of all the files that you want to save.

     c. Purge the remaining files by issuing:

        purge punch all
        purge reader all

  2. Spool your virtual punch to your virtual reader by issuing:

     spool punch * nohold nocont

3. Build the new CP nucleus by issuing:

   vmfload *yourlist* dmkspxx

   where:

   yourlist is the loadlist that matches your system configuration.

   xx is:

t	for UP generated systems
at	for AP generated systems
mt	for MP generated systems

   VMFLOAD punches a new nucleus using the appropriate supplied text files for DMKCPI, DMKFRE, and DMKFRT, which contain the CP FRET Trap code, and places the nucleus in your reader.

- To load the new CP nucleus, issue the following commands:

  ipl 00c
  #cp shutdown reipl

Procedure to Remove CP FRET Trap

- Make sure that you have a backup copy of your current system.
- In addition to the 295, 194, and 294 minidisks, you (MAINT) should have access to the CMS system disks. The search order is:

295	A	R/W	Local modifications minidisk
294	B/A	R/O	CP, PTFs, auxfiles, updates, updated text files, and MACLIBs from the latest PUT
194	C/A	R/O	CP text files and MACLIBs
190	S	R/O	CMS system minidisk
193	T	R/W	CMS base minidisk

- Build a new CP nucleus using the loadlist and control file that matches your system configuration **when CP FRET Trap is not installed**:

  1. Make sure that your virtual card reader and punch do not contain any files:

     a. Close any active files in these devices by issuing:

        close punch
        close reader

     b. Empty your reader and punch of all the files that you want to save.

c. Purge the remaining files by issuing:

purge punch all
purge reader all

2. Spool your virtual punch to your virtual reader by issuing:

spool punch * nohold nocont

3. Build the new CP nucleus by issuing:

vmfload *yourlist yourown*

where:

yourlist is the loadlist that matches your system configuration.

yourown is the control file (DMKSP, DMKSPA, or DMKSPM) used for your system when CP FRET trap is not installed.

VMFLOAD punches a new nucleus using the appropriate text files for DMKCPI, DMKFRE, and DMKFRT, which do not contain the CP FRET Trap code, and places the nucleus in your reader.

4. To load the new CP nucleus, issue the following commands:

ipl 00c
#cp shutdown reipl

Considerations for Users with Local Modifications to CP

Note: If you have local modifications to CP, it is assumed that you are familiar with the procedures for applying them (see "Apply Local and IBM Corrective Updates to Source Code").

- Create a new control file from a copy of your local control file. This new control file requires modifications for CP FRET Trap:

  1. Modify the MACS record in the new local CP FRET Trap installation control file to include an entry for the appropriate MACLIB (DMKSPT for UP, DMKSPAT for AP, or DMKSPMT for MP). This entry must be the first MACLIB named in the MACS record library list.

  2. Insert as the first and second records following the MACS record one of the following sets:

     - For UP systems:

FTL UPDTFT	used by the VMFLOAD and VMFASM commands if you have local modifications to DMKCPI, DMKFRE, and/or DMKFRT
FT UPDTFT	used by the VMFLOAD command if you have no local modifications to DMKCPI, DMKFRE, and/or DMKFRT

     - For AP and MP systems:

APFTL UPDTFT	used by the VMFLOAD and VMFASM commands if you have local modifications to DMKCPI, DMKFRE, and/or DMKFRT
APFT UPDTFT	used by the VMFLOAD command if you have no local modifications to DMKCPI, DMKFRE, and/or DMKFRT

- If you **DO NOT** have local modifications to modules DMKCPI, DMKFRE, or DMKFRT, you can use the "Manual Installation Procedure" on page D-5, substituting the name of your new local CP FRET Trap installation control file in the VMFLOAD command. The appropriate supplied text files for DMKCPI, DMKFRE, and DMKFRT, which include the conditionally (UP, AP, or MP) assembled trap code, are included in the nucleus produced by VMFLOAD.

- If you **DO** have local modifications to modules DMKCPI, DMKFRE, and/or DMKFRT, you need to reassemble the modules you have changed:

  1. Add minidisk 394 (CP source) as D/A to the access search order described in "Manual Installation Procedure" on page D-5.

  2. Reassemble DMKCPI, DMKFRE, and/or DMKFRT (as appropriate) using your new local CP FRET Trap installation control file in the VMFASM command.

     The new text files, which include your modifications and the CP FRET Trap code, are put on the MAINT 295 minidisk. These text files have a filetype of TXTFTL for UP systems and TXTAPFTL for AP or MP systems.

  3. Follow the "Manual Installation Procedure" on page D-5, substituting the name of your new local CP FRET Trap installation control file in the VMFLOAD command.

- To remove the CP FRET Trap, use the "Procedure to Remove CP FRET Trap" on page D-6, substituting in the VMFLOAD command the name of the local control file that you normally use when CP FRET Trap is not installed.

Virtual=Real

Overview

The entry for this option in the sample SPGEN PROFILE is **no**.

The Virtual=Real option allows you to improve the performance of a specified virtual machine by removing the need for paging operations for that machine. If you generate your system with this option, you define an area in real storage as the virtual=real 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 virtual=real area (containing the entire virtual=real virtual machine), and the remaining pageable storage requirements of CP and the other virtual machines.
- You can define only one virtual=real area.
- Only one virtual machine can occupy the virtual=real area at one time.
- A CP nucleus that defines a virtual=real area can be loaded into a real machine (first level installation) or into a virtual machine (second level installation).
- Since this option removes pages from the dynamic paging area, it may degrade the performance of the other virtual machines in the system.

Procedure

- Edit SPGEN PROFILE using the System Product Editor (XEDIT):
  - Change the "CP_vr" entry to **yes**.
  - Because the Virtual=Real option is not supported in a Small CP system, make sure that the "CP_small" entry is **no**.
  - If you want to use the CP FRET Trap option, change the "CP_trap" entry to **yes**. See "Small CP" on page D-2 for more information.
- Modify the system directory (VMUSERS DIRECT) to specify the VIRT=REAL option in the entries of all virtual machines that you want to have the capability of running in the virtual=real area. For

more information about this option, refer to the *VM/SP Planning Guide and Reference.*

- Issue the DIRECT command to place the modified directory on line.
- Specify the amount of storage that you want to reserve for the virtual=real 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 virtual=real area. You must enter the size in K.

*nnnn*K **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 (Chapter 3 or Chapter 4) that you have been using to do the installation.
  - If you are rebuilding the CP nucleus in an existing system, go to "Building the New CP Nucleus" on page K-1.

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 3 or later
- You explicitly control your own disk layout for CP-owned 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.

You can not use this procedure if you are using the Starter System to migrate your VM/SP system, or if your system residence volume is on a 3380-AE4 or 3380-BE4 DASD. In that case, use the SPTAPE command to save the spool files on tape. For more information about SPTAPE, refer to the *VM/SP CP Command Reference.*

In the SHUTDOWN/WARM IPL procedure, the new VM/SP system is WARMstart 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 may fail and a COLD START IPL may be required.

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 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 System" on page 4-1, with the following additions:

  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 SP5RES, 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,
         SP5RES,
         PACKCC
SYSRES   SYSVOL=SP5RES,
         SYSTYPE=3380,
         SYSWRM=(10,10,SP4RES),
         SYSCKP=(20,10,SP4RES),
```

The order of the new volumes (SP5RES 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 (such as VMUSERS DIRECT) to the first level userid (such as MAINT) 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 userid 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 may contain spool files). DO NOT destroy the SYSWRM and SYSCKP spaces on the first level system residence volume.

- Log on to the first level userid 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, issue the DIRECT command to process the tailored CP directory and write it to the DASD volume defined for the new system. This enables you to log on to the new system after the shutdown.

- SHUTDOWN 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 page 15-72.

- Continue the migration procedure with "Step 8. Load and Save HELP (Optional)" on page 4-36.

Appendix F. Enlarging the CMS Nucleus To Contain the Y-Disk Directory (Y-STAT)

Overview

Since CMS is designed to run as a shared system, performance is generally improved when directories for the S- and Y-disks (the S-STAT and Y-STAT) are included in the CMS nucleus.

When IPLing 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-disk 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-disks.

   Enter:

   query disk

   and record the number of files on both disks.

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.

3. Compute the additional amount of storage needed (*A*) by using the applicable formula:

   $A = 40 * F - D$ (for 800-byte formatted disk)

   - OR -

   $A = 64 * F - D$ (for all others)

   where *F* is the number of files on S- and Y-disks, 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	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 may need to make an adjustment to the next lower segment to compensate. Also, you may 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.

  Issue the following commands:

  xedit cmsload exec
  top
  c/ SLC LF00000 / SLC L*xxxxxx* / *
  file

  where *xxxxxx* is the new CMS starting address (NUCALPHA) that you computed above.

- Modify the CMS entry in the DMKSNT file.

  Enter:

  xedit dmksnt assemble

  and change the following four parameters on the CMS NAMESYS macro:

  Note: To make these modifications, you may have to reposition other 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
  spgen assemble dmksnt

- Create a new SLC L*xxxxxx* file that is referenced by CMS.

  Issue the following commands:

  access 193 b/a
  copyfile SLC LF00000 b = L*xxxxxx* a
  xedit SLC L*xxxxxx* a
  top
  c / F00000 / *xxxxxx* / *
  file

  where *xxxxxx* is the new CMS starting address (NUCALPHA) that you previously computed.

- Rebuild the CP nucleus:

  access 295 d
  spgen nucleus cp

- Shut down and reIPL to make the changes effective:

  #cp shutdown reipl

  Note: Since this command performs an automatic warm start that automatically logs you on to the OPERATOR userid, you may want to issue the command from the system operator's console.

- Log on as MAINT.

- Build and save the CMS nucleus:

  Note: Before you build the nucleus, you may want to re-examine the CMS nucleus generation profile (DMSNGP ASSEMBLE) to make sure that the responses are defined correctly for this nucleus. If you change DMSNGP, issue **spgen assemble dmsngp cms** *to reassemble the file.*

  spgen nucleus cms

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 239-255, which are page numbers 3824-4095, and storage location X'F00000'-X'FFFFFF'. This 16 Meg 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 4 Meg. It requires 256 bytes of real storage for each CMS user. The CMS saved system is contained in segments 64-79, which are pages 1024-1279, and storage location X'400000'-X'4FFFFF'.

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 disk:

  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 LF00000" 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 may 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 LF00000 / SLC L400000 /*
  top
  change / SLC L000000 / SLC L500000 /*
  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 L500000 a
input $SLC   500000      (there must be two blanks between SLC and
                          the address)
set hex on
change /$/X'02'/         (X'02' is an unprintable loader control
                          character)
file
```

- Modify the SYSPGNM and SYSHRSG parameters in the NAMESYS macro for CMS in DMKSNT.

  The following figures show a sample (3380 DASD) NAMESYS entry for CMS, and how the revised NAMESYS entry might look.

```
*********************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT 239 = F00000                     *
*     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 8, PAGE 3     (303 PAGES)            *
*       302 PAGES FOR CMS, 1 FOR CP INFORMATION.                    *
*********************************************************************
CMS       NAMESYS    SYSNAME=CMS,                                   X
               SYSVOL=VMSRES,                                       X
               SYSSTRT=(006,1),                                     X
               SYSPGNM=(0-8,14-34,3824-4095),                       X
               SYSPGCT=302,                                         X
               SYSHRSG=(239-255),                                   X
               SYSSIZE=256K,                                        X
               VSYSADR=190,                                         X
               SYSCYL=487,                                          X
               PARMRGS=(0,15),                                      X
               VSYSRES=VMSRES
          EJECT
```

Figure G-1. Sample NAMESYS Entry for CMS.

```
*********************************************************************
*     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 8, PAGE 3     (303 PAGES)            *
*       302 PAGES FOR CMS, 1 FOR CP INFORMATION.                    *
*********************************************************************
CMS       NAMESYS    SYSNAME=CMS,                                   X
               SYSVOL=VMSRES,                                       X
               SYSSTRT=(006,1),                                     X
               SYSPGNM=(0-8,14-34,1024-1279),                       X
               SYSPGCT=302,                                         X
               SYSHRSG=(64-79),                                     X
               SYSSIZE=256K,                                        X
               VSYSADR=190,                                         X
               SYSCYL=487,                                          X
               PARMRGS=(0,15),                                      X
               VSYSRES=VMSRES
          EJECT
```

Figure G-2. Example of Alternate CMS Nucleus Placement

- Assemble DMKSNT:

  spgen assemble dmksnt

- Rebuild the CP nucleus:

  spgen nucleus cp

- Shut down and reIPL to make the changes effective:

  #cp shutdown reipl

  Note: Since this command performs an automatic warm start that automatically logs you on to the OPERATOR userid, you may want to issue the command from the system operator's console.

- Log on as MAINT.

- Build and save the CMS nucleus:

  Note: Before you build the nucleus, you may want to re-examine the CMS Nucleus Generation Profile (DMSNGP ASSEMBLE) to make sure that the responses defined in the DEFNUC macro are correct for this nucleus. If you change DMSNGP, issue **spgen assemble dmsngp cms** *to reassemble the file. For more information about the DEFNUC macro, refer to the VM/SP Planning Guide and Reference.*

  spgen nucleus cms

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. Refer to 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 reIPL to make the changes effective.
- Build and save the GCS nucleus.

Procedure

This procedure demonstrates how to relocate your GCS saved system in a 10 M virtual machine. Originally 1 M in size and loaded at X'400000', the segment is moved to X'800000' and increased in size to 2 M.

- Log on as MAINT and access the GCS system disk:

  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 may 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 NAMESYS macro for GCS in DMKSNT. As a result, you may 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 to decimal.

       c. Subtract one; this gives you the last page of the nucleus (in this case, the last page is 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 518 (pages 0-5 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) NAMESYS entry for GCS, and how the revised NAMESYS entry might look.

```
*************************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT 128 = 800000                         *
*     THE SPACE FOR GCS IS ALLOCATED ON VMPK01, AS FOLLOWS:             *
*         CYL 1, PAGE 1   TO CYL 3, PAGE 23    (263 PAGES)              *
*************************************************************************
GCS      NAMESYS SYSNAME=GCS,                                           X
               SYSVOL=VMPK01,                                           X
               SYSSTRT=(01,001),                                        X
               SYSPGNM=(0-5,2048-2559),                                 X
               SYSPGCT=518,                                             X
               SYSHRSG=(128-159),                                       X
               SYSSIZE=256K,                                            X
               VSYSADR=595,                                             X
               SYSCYL=790,                                              X
               VSYSRES=VMPK01,                                          X
               PROTECT=OFF,                                             X
               VMGROUP=YES
         EJECT
```

Figure H-1. Sample NAMESYS Entry for GCS

```
*************************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT  64 = 400000                         *
*     THE SPACE FOR GCS IS ALLOCATED ON VMPK01, AS FOLLOWS:             *
*         CYL 1, PAGE 1   TO CYL 3, PAGE 23    (263 PAGES)              *
*************************************************************************
GCS      NAMESYS SYSNAME=GCS,                                           X
               SYSVOL=VMPK01,                                           X
               SYSSTRT=(01,001),                                        X
               SYSPGNM=(0-5,1024-1279),                                 X
               SYSPGCT=262,                                             X
               SYSHRSG=(064-079),                                       X
               SYSSIZE=256K,                                            X
               VSYSADR=595,                                             X
               SYSCYL=790,                                              X
               VSYSRES=VMPK01,                                          X
               PROTECT=OFF,                                             X
               VMGROUP=YES
         EJECT
```

Figure H-2. Example of Alternate GCS Nucleus Placement

- Assemble DMKSNT:

  access 191 a
  access 194 c
  spgen assemble dmksnt

- Rebuild the CP nucleus:

  spgen nucleus cp

- Shut down and reIPL to make the changes effective:

  #cp shutdown reipl

  Note: Since this command performs an automatic warm start that automatically logs you on to the OPERATOR userid, you may want to issue the command from the system operator's console.

- Log on as MAINT.

- Build and save the GCS nucleus:

  access 595 c

  Note: Before you build the nucleus, you may want to create a new GCS configuration file for this nucleus. See the GCS step in the installation procedure.

  itask build gcs *systemname*

  *Note: If you do not specify a **systemname**, the default is GCS.*

  ipl cms

Appendix I. 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.

Note: 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 disk where that library resides.

DOSMAC creates a DSERV listing on your CMS disk 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 filename that you specify. If you do not specify a filename, the default is DOSMAC.

If you have too many VSE macros to move to your CMS disk, 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 filename (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	IOTAB	SGPMAIN
CDMOD	ISMOD	SGPSUB
DAMOD	MCRAS	SGSVC
DAMODV	MTMOD	COBBG
FOPT	SGCCWT	COBF2
IOINTER	SGEND	

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

- 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 disk.
*   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 may take a substantial length of time, up to several hours for a large macro library. A detailed description of the ESERV command can be found in the VM/SP CMS Command Reference. For more information on how to use the ESERV command, refer to the VM/SP CMS for System Programming. Refer to the Guide to the DOS/VSE Assembler, GC33-4024 for a detailed description of the VSE ESERV control statements.

Appendix J. Installing the 3704/3705 Control Program

Note: Do not use these instructions if you are installing SNA products such as NCP Version 3 for the 3705 or NCP Version 3 for the 3725 (5667-124), and Emulation Program (EP) for the 3705 or EP for the 3725 Release 2 (5735-XXB). Refer to the product specific manuals for instructions on loading the 3705/3725.

This section describes generating and loading only the 3704 and 3705 communication controller through VM/SP commands. VM does not provide loading or dumping facilities for a 3725 communication controller. The 3725 requires ACF/SSP V2/3725. (SSP V2/3725 also supports the IBM 3705 for loading and dumping.)

For more information about how ACF/SSP handles a 3725 controller and a 3705, refer to the following manuals.

- *EP.3725 Installation and Resource Definition Guide and Reference*, SC30-3172.

  This manual explains how to generate and load the EP for a 3725.
- *EP.3705 Generation and Utilities Guide and Reference*, GC30-3242.

  This manual explains how to generate and load the EP for a 3705.
- *ACF.NCP-SSP, V3 Installation and Resource Definition Guide*, SC30-3253.

  This manual explains how to generate and load the NCP.
- *ACF.NCP V4, ACF/SSP V3 Diagnosis Guide*, SC30-3255.

  This manual explains how to dump the contents of the 3705 or 3725 and how to run ACF/TAP and CRP.

Several commands and EXEC procedures generate and load the 3704/3705 control program. These commands and EXEC procedures are executed in a CMS virtual machine. The commands are part of the VM/SP system and are distributed with it.

A special version of the IBM 3704/3705 Emulation Program Support Package for OS/VS, Order No. 5744-AN1 EP/5735-XXB SCP is available from PID for use under VM/SP. This version of the 3704/3705 package contains two CMS EXEC procedures for generating and loading the 3704/3705 control programs using VM/SP.

This chapter describes the step-by-step procedure for generating and loading the 3704/3705 control program. Each EXEC procedure and command is described as it is used. The action required at each step is summarized first and then explained in detail. "Part 1. Planning for System Generation" of the *VM/SP Planning*

Guide and Reference lists all the documentation, physical devices, programming, and other materials you need before starting to generate the 3704/3705 control program.

Note: The instructions that follow are specific to the EP/VS tape that is in OS format. If you are using the EP/VS product that is shipped on the system offering stacked tape, refer to the instructions on the Memo to Users.

Step 1. Log on the VM/SP System

VM/SP supports the EP (Emulation Program) type of control program. The VM/SP system that you load also must have been generated with:

- The IBM 3704 or 3705 Communications Controllers specified on a RDEVICE system generation macro in DMKRIO ASSEMBLE.
- The NAMENCP macro coded to create an entry in the VM/SP system name table (DMKSNT) for the 3704/3705 control program.
- Space reserved on a CP-owned volume to contain a copy of the 3704/3705 control program.

These VM/SP system generation requirements are described in the *VM/SP Planning Guide and Reference.*

Step 2. Set Up a CMS Virtual Machine

You must IPL CMS in a virtual machine and be sure that the necessary devices are attached.

The 3704/3705 control program is generated using commands and EXEC procedures that execute in a CMS virtual machine. The CMS virtual machine must have the following resources:

- At least 1024K (1M) of virtual storage.[1]
- One tape drive (9 track, 1600 or 6250 bpi).

---

[1] Depending on the size and options of the EP Program you build, and depending on the blocksize and track size of DASD you use, you may have to increase the minimum amount of virtual storage. Verify that the 3705 assembler has been built; then use a larger virtual storage size, e.g., 4096K or greater to continue.

- Space available on the CMS A-disk:

  120 cylinders of a 3330 disk
  203 cylinders of a 2314 disk
  300 cylinders of a 3340 disk
  60 cylinders of a 3350 or 3380 disk
  80 cylinders of a 3375 disk
  7200 pages of an FB-512 disk.

If the CMS virtual machine does not have these resources, use the CP DEFINE command to redefine the size of your virtual storage or send a message to the operator requesting the needed tape or disk device be attached.

Be sure that there are no files on the A-disk with a filetype of COPY or TEXT. Use the CMS RENAME command to temporarily change such filetypes. A naming conflict can end the installation procedure for the the distribution tape.

You need CP command privilege classes A, B, or C to install the 3704/3705 control program. Check with the system administrator to ensure that your VM/SP directory entry has the appropriate command privileges.

Step 3. Load the IBM 3704/3705 Control Program Distribution Tape

In this step you load the files onto a CMS disk that you access as filemode A.

Note: **Do not** *access the CMS System-Disk (190) as filemode A.*

Use CMS commands to position the distribution tape at the proper file and to create CMS disk files from the tape files. The first file created from the tape files is an EXEC procedure that processes the rest of the tape files and creates the CMS disk files.

Mount and attach the distribution tape to your virtual machine. The distribution tape contains ten physical files, separated by tape marks. The tenth file contains the INSTEP and ARNGEND EXEC procedures, which create the necessary CMS files from the other tape files.

Issue the following commands to position the tape at the beginning beginning of the tenth file:

```
tape rew
tape fsf 9
```

Next, issue the following command to create the INSTEP EXEC A1 and ARNGEND EXEC A1 files from the tenth file:

```
tappds * exec
```

If the files are successfully created, you receive the following responses:

```
FILE 'ARNGEND EXEC A1' COPIED
FILE 'DMSARD  EXEC A1' COPIED[2]
FILE 'DMSARX  EXEC A1' COPIED[2]
FILE 'DMSGRN  EXEC A1' COPIED[2]
FILE 'DMSTMA  EXEC A1' COPIED[2]
FILE 'INSTEP  EXEC A1' COPIED
```

Step 4A. Invoke INSTEP EXEC

Before you invoke the INSTEP EXEC procedure:

1. XEDIT INSTEP EXEC, deleting the RENAME commands for DMSARD, DMSARX, DMSGRN, and DMSTMA.

2. Erase the DMSARD, DMSARX, DMSGRN, and DMSTMA EXEC files.

3. Access the CMS System-Disk (190) as T/A to get access to the mode 1 files.

To invoke the INSTEP to load all the necessary files and generate the 3705 Assembler, enter:

```
instep
```

The INSTEP EXEC procedure generates the 3705 Assembler and creates the macro and text libraries that are needed to generate a 3704/3705 control program. The INSTEP EXEC procedure issues the following informational messages:

```
REWIND COMPLETE
BUILD STAGE ONE MACLIB
LOADING 'GEN3705 MACLIB'
```

INSTEP uses the third tape file to create the CMS file GEN3705 MACLIB A1. INSTEP also issues the messages:

```
BUILD STAGE TWO MACLIBS
LOADING 'MAC3705 MACLIB'
```

using the fifth tape file to create the CMS file MAC3705 MACLIB A1. Using the sixth tape file, INSTEP creates the CMS file OBJ3705 MACLIB A1, and issues the messages:

```
BUILD STAGE TWO TXTLIB
```

```
LOADING 'OBJ3705 MACLIB'
RENAME OBJ3705 MACLIB A1 OBJ3705 TXTLIB A1
```

---

[2] These files may not be present on later releases of the EP/5735-XXB distribution tape.

Finally, INSTEP issues the message:

```
LOAD 3705 ASSEMBLR FILES
```

and loads the assembler text files from tape via the TAPPDS command. The files copied are listed off in messages in the form:

```
FILE 'fn EPTAPE A1' COPIED
```

INSTEP invokes the ARNGEND EXEC procedure to generate the 3705 Assembler, after issuing the message:

```
BUILD 3705 ASSEMBLR MODULES.
```

Step 4B. Invoke ARNGEND EXEC

Note: If you are a licensed program user entering the procedure at this step, you must issue the ARNGEND command to manually invoke the ARNGEND EXEC procedure.

The ARNGEND EXEC procedure displays the following status and error messages:

```
ENTER TARGET DISK MODE FOR 3705 ASSEMBLR MODULES
DEFAULTS TO S-DISK IF NONE ENTERED
```

You enter the mode letter of the disk that contains the 3705 assembler modules when the assembler is used. This may be a different disk than the one on which the module now reside. If you enter a mode letter, ARNGEND uses that mode letter as the "targetmode" operand of the GENDIRT command when it creates the auxiliary directory for the 3705 assembler. If you do not specify a mode letter, GENDIRT assumes a mode of S.

If the 3705 assembler text files are not loaded successfully, or if the assembler generation procedure fails, the following message appears:

```
ASM3705 GEND FAILED
```

When the last message:

```
END OF EPTAPE INSTALL
```

appears on the terminal, you no longer need the distribution tape. At this time, the 3705 Assembler program, the macro libraries for the Stage 1 and Stage 2 generation procedures, and the text library for the Stage 2 generation procedure all exist on the CMS A-disk.

Note: You may find it helpful to dump the contents of the A-disk to tape at this time. If you save the tape dump, you have the pre-Stage 1 files. If errors are later found, you may need these files.

Step 5. Code the 3704/3705 Control Program Macro Instructions

Code the 3704/3705 control program macro instructions and place them in a CMS file. Use the System Product Editor to create the file, which must have a filetype of ASM3705. VM/SP recommends that you assign the same filename to this CMS file as you specified previously in the NAMENCP macro. If you specify the SAVE option on the GEN3705 command, the filename must be the same. Stage 1 of the 3704/3705 control program generation procedure uses the ASM3705 file as input.

Use the *3704 and 3705 Generation and Utilities Guide* to code the macro instructions. Follow the macro instruction formats described in that publication except where suggestions and requirements are indicated in the following paragraphs.

BUILD Macro Instruction

The BUILD macro must be the first macro in the CMS file. Figure J-1 lists the operands which VM/SP requires, recommends, or does not support. For all other operands, refer to the *3704 and 3705 Generation and Utilities Guide*.

Operand	Comments
LOADLIB=dsname **OBJLIB=dsname**	Required by the BUILD macro, but does not apply to VM/SP. Specify a valid dsname.
JOBCARD= { **YES** / **NO** }	VM/SP recommends JOBCARD=YES for EP.
NEWNAME= { **NCP001** / **PEP001** / **VMEP01** / **symbol** }	VM/SP requires that the value of NEWNAME be the same as the name previously specified in the NAMENCP macro and the name that you subsequently specify in the SAVENCP command. Also, if you issue the GEN3705 command with the SAVE option, the value of NEWNAME must be the same as the "fname" you specify on the GEN3705 command.
QUALIFY= { **symbol** / **NONE** / **SYS1** }	VM/SP requires the default value.
UT1=dsname **UT2=dsname** **UT3=dsname**	VM/SP ignores these operands.

Figure J-1. BUILD Macro Operands for VM/SP

CSB Macro Instruction

The CSB macro instruction is required. See the *3704 and 3705 Generation and Utilities Guide* for more information about coding the CSB macro instruction.

GROUP and LINE Macro Instructions

These macros describe the physical and logical configuration of the communications network accessed through the 3704/3705 control program. Since VM/SP does not support either multi-drop lines or cluster control units, the 3704/3705 configuration for VM/SP is generally simple, with only one GROUP macro for each communications scanner. For VM/SP, it is often easiest to specify most of the operands of the LINE macro on the GROUP macro. The *3704 and 3705 Generation and Utilities Guide* describes the GROUP and LINE macro instructions in detail and lists all the operands of the configuration macros, telling you where each operand is described and also where it may be coded.

VM/SP requires the DUPLEX and FEATURE operands. These operands allow VM/SP to detect and respond to a terminal attention interrupt and to recognize when a data set has been hung up. For the GROUP macro, VM/SP requires the default value for the REPLYTO operands and recommends the default value for the TEXTTO operand.

GENEND Macro Instruction

The GENEND macro indicates the end of the 3704/3705 macro input file. It must be the last macro in the CMS file you are building as input to Stage 1.

Special Macro Coding Considerations for the Emulation Program (EP)

There are no strict dependencies between the host access method and the emulation program; consequently, few guidelines are necessary for an emulation program generation.

- Be careful when configuring emulator lines for CPT-TWX terminals. While VM/SP normally accepts incoming calls from either 1050 or 2741 terminals on the same physical line, that same line cannot be used for CPT-TWX terminals.

- When generating the VM/SP system, ensure that the hardware configuration specified in the CP module DMKRIO matches the configuration of the Emulation Program for CPT-TWX lines; the exact configuration of 1050 and 2741 lines is not critical.

Note: The base address of the 3704/3705 (the address used to load and/or dump the control program) can never be specified for use as a telecommunications line. VM/SP treats the base address as a separate entity for use only during the load and dump operation.

Step 6. Define the Macro and Text Libraries

The macro and text libraries created from the distribution tape in Step 3 must be made available to CMS. One macro library (GEN3705) is needed for the Stage 1 assembly procedure and one macro library (MAC3705) and one text library (OBJ3705) are needed for the Stage 2 generation procedure. It is easiest to define all the libraries before starting Stage 1. Issue the following CMS GLOBAL commands:

```
global maclib gen3705 mac3705
global txtlib obj3705
```

Note: There is a MACLIB (PTF3705) which contains all the corrected macros. For this MACLIB to be effective, you must specify it first in the GLOBAL MACLIB search.

Step 7. Complete the Stage 1 Assembly Procedure

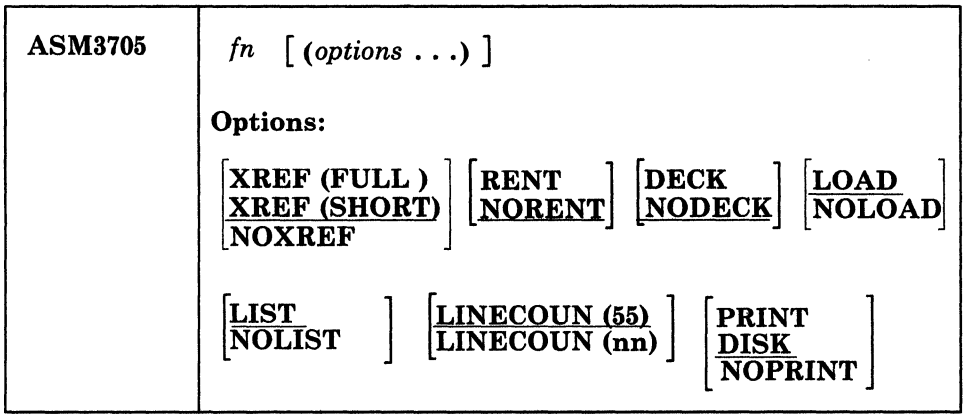
The Stage 1 assembly procedure accepts the CMS file you created in "Step 5. Code the 3704/3705 Control Program Macro Instructions" on page J-6 as input and produces the Stage 2 input file that you need for "Step 8. Complete the Stage 2 Generation Procedure" on page J-12.

Complete the Stage 1 assembly procedure by invoking the 3705 Assembler to process the 3704/3705 control program macro instructions. It produces one file with the same filename as the input file and with a filetype of TEXT. This TEXT file contains 3705 Assembler source statements and job control language (JCL) statements.

The ASM3705 Command

Use the CMS ASM3705 command to invoke the 3705 Assembler to assemble the macro instruction file. The options you select control the 3705 assembler processing and output.

The format of the ASM3705 command is:



where:

fn
specifies the filename of the source file to be assembled. This source file contains the 3704/3705 control program macro instructions. The file must have a filetype of ASM3705 and fixed-length, 80-character records.

Options:

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 on disk for the program that was assembled.

NOLOAD
suppresses the creation of a TEXT file on disk 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
puts the LISTING file on disk.

NOPRINT
produces no LISTING file.

LINECOUN nn
specifies the number of lines per output printer page. A default of 55 lines is assumed.

Note: All of the options of the 3705 XF Assembler are supported and may be used with the ASM3705 command, with the exception of ALIGN/NOALIGN and TEST/NOTEST.

Files Created by the ASM3705 Command:

TEMPORARY WORKFILES: Three files are temporarily created for each assembly:

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 disk 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: One or two permanent files may be created during a successful assembly:

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 disk from which the source file was read,
2. Its parent, or
3. The primary disk, 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.

Special Considerations for the Stage 1 Assembly

The Stage 1 assembly 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.

The LISTING file produced by the Stage 1 assembly is quite large. If you let the ASM3705 command option default to DISK, much of the space on your A-disk is used. Therefore, VM/SP 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. VM/SP 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.

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

Step 8. Complete the Stage 2 Generation Procedure

During the Stage 2 generation procedure the TEXT file produced in "Step 7. Complete the Stage 1 Assembly Procedure" on page J-8 is scanned. That TEXT file contains several job steps of 3705 assembler source statements with embedded OS JCL statements. The JCL statements are removed and a unique CMS 3705 assembler source file is created for each job step in the input file. An EXEC procedure is also created to assemble and link edit the source files. When you invoke the EXEC procedure, it produces the load module file (and, optionally, saves a copy of the control program in page-format on a CP-owned volume).

The GEN3705 Command

Use the CMS GEN3705 command to invoke the Stage 2 service program. 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 disk, or if GEN3705 invokes the EXEC procedure automatically.

The format of the GEN3705 command is:

GEN3705	*fn ft [fm] [(options ...)]* *Options:* [**RUN** / **NORUN**] [**SAVE** / **NOSAVE**]

where:

fn
: specifies the filename of the Stage 2 input stream produced by the Stage 1 assembly. The file must contain fixed-length, 80-character records.

ft
: specifies the filetype of the Stage 2 input stream. The filetype is normally TEXT.

fm
: specifies the filemode.

Options:

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 specify 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.

```
fname00  ASM3705      fnameL0  TEXT       fname EXEC
fname01  ASM3705      fnameL0  TEXT
  .      .     .      .
  .      .     .      .
  .      .     .      .
fnamenn  ASM3705      fnameLn  TEXT
```

A separate ASM3705 file is created for each assembly job step in the Stage 2 input file. Each ASM3705 file created by GEN3705 is given a unique filename of the form 'fnamenn'. The first six characters of the input filename 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 Stage 2 EXEC procedure.

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 filename of the form 'fnameLn'. The first six characters of the input filename 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 filenames assigned to the linkage editor and assembler files must be different. If the filenames 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 contains use of 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 filename of the Stage 2 input file is used as the 'ncpname' operand for the SAVENCP command.

Special Considerations for the Stage 2 Generation Procedure

VM/SP recommends that you specify the RUN option. When the RUN option is specified, 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.

If you do not specify the SAVE option, you have to explicitly issue the SAVENCP command. If you do specify the SAVE option, use of be sure that the input file has the same filename 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.*

Step 9. Invoke the EXEC Procedure Created by GEN3705

If you specified RUN on the GEN3705 command, this step is executed for you. If you did not specify RUN on the GEN3705 command, you must invoke the EXEC procedure that the GEN3705 program created.

The EXEC procedure is given the same filename as the GEN3705 input file. It is invoked by entering that filename at the terminal. For example, if the input file is NCP320 TEXT, the EXEC file is named NCP320 EXEC, and can be invoked by issuing:

ncp320

at the terminal.

This 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).

You need not issue the ASM3705 and LKED commands that create the 3704/3705 control program load module; the EXEC procedure does that for you. The ASM3705 command is described in Step 6. The FILEDEF, LKED, and TXTLIB commands are described in the *VM/SP CMS Command Reference.*

Step 10. Save the 3704/3705 Control Program Image on Disk

If you specified SAVE on the GEN3705 command, this step is executed for you. If you did not specify SAVE on the GEN3705 command, you must issue the SAVENCP command yourself.

Note: The VM/SP command privilege class A, B, or C is required to use the SAVENCP command.

The SAVENCP Command

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.

The format of the SAVENCP command is:

SAVENCP	*fn* [(*options* ...)] *Options:* [**ENTRY** *symbol* / **CXFINIT**] [**NAME** *ncpname* / *fn*] [**LIBE** *libraryname* / *fn*] [**CAMOD** {**0** / **1**}]

where:

fn
is the filename of the LOADLIB file where the 3704/3705 control program load module resides; unless LIBE is specified, in which case, it specifies 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 Stage 2 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 *librarynname*
is the filename of a load module library file, filetype LOADLIB, which contains the control program image as member 'fname'.

CAMOD $\left\{\begin{matrix}0\\1\end{matrix}\right\}$
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.

Step 11. Load the 3704/3705 Control Program

When you do CP LOADING EP in a twin-tail environment, you should never code the CPNAME operand and the RDEVICE macro.

The 3704/3705 control program is automatically loaded each time the VM/SP system is loaded, if the CPNAME operand was specified on the RDEVICE macro when VM/SP was generated and if the 3704/3705 is online. If the CPNAME operand was not coded, you must issue the CP NETWORK LOAD command line to load a 3704/3705 control program into the 3704/3705 Communications Controllers' storage.

The NETWORK LOAD Command

Use the NETWORK LOAD command to initiate the loading of an EP control program into a 3704/3705 Communications Controller.

The format of the NETWORK LOAD command line is:

NETwork	*LOAD raddr ncpname*

where:

LOAD
initiates the control program load operation.

raddr
is the real address of the 3704/3705 to be loaded.

ncpname
is the name, defined by a NAMENCP macro, of the 3704/3705 control program image to be loaded into the 3704/3705 specified by raddr.

How the NETWORK LOAD Command Works: The NETWORK LOAD command accesses the control program image using the information in the system name table (DMKSNT) entry created by the NAMENCP macro. If the 3704/3705 specified in the command is not in an "IPL Required" state at the time the command is issued, the message:

```
DMKNET461R CTLR vdev IPL NOT REQUIRED; ENTER "YES" TO CONTINUE:
```

appears at the terminal. If the reply to the message is other than "yes," the command terminates without loading the 3704/3705. Otherwise, the loader bootstrap routines are written to the 3704/3705 and loading starts. VM/SP does not execute the "bring-up" test routines as a part of the load process. If these tests are to be made, they must be run from a virtual machine with the 3704/3705 dedicated.

When the load of the control program image is complete, the command processor verifies that the 3704/3705 configuration described by the control program can be serviced by the VM/SP CP control blocks in storage.

Special Considerations for Loading the 3704/3705 Emulation Program

If a 3704/3705 Emulation Program is automatically reloaded after a 3704/3705 failure, the system may loop after the restart. The message:

```
DMKRNH463I CTLR rdev unit check; restart in progress
```

and two responses:

```
CTLR rdev DUMP COMPLETE
CTLR rdev ncpname LOAD COMPLETE
```

indicate that the 3704/3705 has been reloaded. If the system loops after the second response, you must reset all emulator lines from the 3704/3705 control panel.

If the automatic dump feature is not enabled, one of the following messages:

```
DMKRNH462I CTLR rdev unit check; IPL required
```

-- or --

```
DMKRNH464I CTLR rdev CC=3;  press3705 'LOAD' button
```

indicates a 3704/3705 abnormal termination. The 3704/3705 Emulation Program must be reloaded via the NETWORK LOAD command. If the system loops when an attempt is made to enable the lines, you must reset all emulator lines from the 3704/3705 control panel.

The procedure for resetting emulator lines from the 3704/3705 control panel is described in the *Guide to Using the IBM 3704 Communications Controller Control Panel* and the *Guide to Using the IBM 3705 Control Panel.*

Step 12. Log on VM/SP through the 3704/3705

Because a 3704/3705 can support emulator-mode lines and can also support a variety of terminals, the procedure for logging on is sometimes complicated. Use the following procedure to log on to VM/SP.

1. Turn the power on for your terminal and wait 15 to 30 seconds.
2. Look for an online message at your terminal.
   a. If one of the following messages:

      ```
      VM/370 online      xxxxxx xxxxxx
      ```

      -- or --

      ```
      xxxxxx xxxxxx      VM/370 online
      ```

      appears at your terminal, your terminal is a 2741 connected to VM/SP via a 2701/2702/2703 line or via a 3704/3705 line in emulation mode. You can proceed with the normal logon procedure for your type of terminal, as described in the *VM/SP Terminal Reference.*

   b. If the message:

      ```
      VM/370 online
      ```

      appears at your terminal, your terminal is a 1050, 3101, or CPT-TWX Model 33/35 terminal connected to VM/SP in EP mode. You can proceed with the normal logon procedure for your terminal type. This procedure is described in the *VM/SP Operator's Guide.*

Step 13. Apply PTFs to the 3704/3705 Load Library

If necessary, it is possible to apply Program Temporary Fixes (PTFs) directly to the 3704/3705 load library[3] The CMS ZAP program applies the PTF. Refer to "ZAP MODULE" on page 15-98 for more information about the ZAP service program.

[3] PTF3705 MACLIB contains all corrected macros. You must specify this MACLIB first in the GLOBAL MACLIB search.

Step 14. Test the 3704/3705 Control Program

After you have generated a 3704/3705 control program, loaded it, and logged on, you may want to test the 3704/3705 control program. Several CP commands are provided to control the operation, check the status, and dump the contents of the 3704/3705. The NETWORK command loads and dumps any 3704/3705 control program. The existing CP commands (ENABLE, QUERY, DISPLAY, and VARY) also provide support for EP 3704/3705 control programs. The NCPDUMP command formats and prints a dump of 3704/3705 storage. Use these commands to test the 3704/3705 control program.

The NETWORK, ENABLE, DISABLE, QUERY, and VARY commands are described in the *VM/SP CP Command Reference*. The NCPDUMP command is described in the *VM Diagnosis Guide*.

Appendix K. Building CP and CMS Nuclei Using SPGEN

This chapter discusses using SPGEN to:

1. Build a new CP nucleus.
2. Build a new CMS nucleus.

Building the New CP Nucleus

If you have not already done so, logon to MAINT and issue the following commands:

define storage 16m

```
DMKDSP450W CP entered; disable wait PSW '00020000 00000000'
```

ipl cms

ENTER

Before continuing, make sure you have:

- The correct access order. You can verify this by looking in the SPGEN PROFILE. Make sure the minidisk containing the service code is accessed before the minidisk containing the base code.
- Write access to the production CP system residence volume (VMSRES).

Next, issue:

spgen nucleus cp

This spools the punch continuous to MAINT virtual reader and invokes the VMFLOAD command. (VMFLOAD punches, to MAINT's reader, a file containing the CP loader followed by all the text files for the nucleus). It also IPLs your reader.

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

nnnnn is the starting and ending cylinders/blocks for the nucleus. Verify that this nucleus has not expanded into another allocated area such as "TEMP," "WARM START," etc., or unpredictable results may occur.

The message below does not automatically display on many processors. Check your processor reference manual for information about displaying and interpreting the PSW. If the PSW message is not displayed, press PA1 to get a CP READ state, and issue: **#cp display psw**.

You receive the following message:

```
DMKDSP450W CP entered; disabled wait PSW '00020000 00000012'
```

This is the normal return if the nucleus loaded correctly. If the PSW displayed is not "00020000 00000012," refer to the LOADER WAIT state codes in *VM/SP System Messages and Codes*.

From the operator's console logon to OPERATOR and issue the following:

#cp shutdown reipl

This command does 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 userid.

```
DMKCPJ952I nnnnK system storage
```

Based on your processor type, *nnnn*K may be less than your actual system storage.

```
.
.
.
DMKCPI996I Initialization complete
```

We now recommend that you do a full system backup. Refer to "Step 20. Do a Full System Backup (Recommended)" on page 3-122.

This copy may later be restored using the stand-alone version of the DDR program on the real machine.

Building the New CMS Nucleus

If you have not already done so, logon to MAINT. When you prepare to build a new CMS nucleus, be sure that you have access to the 190 system disk, MAINT's 193 minidisk (text decks), and MAINT's 194 minidisk (loadlists and control files). For this procedure you need another disk, 390, that is initially an exact copy of the 190 system disk. If you are unfamiliar with how to create and format this disk, see Steps 1 and 2 of the existing VM system installation procedure (Chapter 4).

In this procedure, we build and test the new nucleus using the 390, create a disk file of this nucleus, use this disk file to create the new nucleus on the system disk (190).

Be sure that your virtual card punch and reader do not have any class M or N files in them and that your virtual punch is spooled to your virtual reader. Enter:

spool reader hold

spool printer *

Routes the nucleus load map to your card reader instead of the virtual printer.

To use the 390 minidisk, you need to modify the DMSNGP ASSEMBLE file. "Sample CMS Nucleus Generation Profile (DMSNGP ASSEMBLE)" on page C-5 contains a sample of the DMSNGP Profile. Using the System Product Editor to edit the file, change:

the line containing: **SYSDISK=190,** to now read: **SYSDISK=?,**.

the line containing: **SAVESYS=YES,** to now read: **SAVESYS=?,**.

the line containing: **IPLADDR=190,** to now read: **IPLADDR=?,**.

Once you have made the changes, file it and reassemble DMSNGP ASSEMBLE. You must also have the disk containing the reassembled file (the DMSNGP TEXT file) accessed.

spgen nucleus cms

This spools the punch continuous to MAINT's virtual reader and invokes the VMFLOAD command. VMFLOAD uses the system-supplied CMSLOAD EXEC and DMSSP CNTRL files and punches, to MAINT's reader, a file containing the CMS loader followed by all the the files for the nucleus. It also IPLs your reader.

Note: Each of the following prompts appears ONLY if the corresponding statement in DMSNGP is missing or empty (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?).

DMSINQ606R System disk address =

390

Specify 390 since this is the minidisk you have defined to contain the new CMS nucleus.

DMSINQ615R Y-disk address =

ENTER

19E is the default.

DMSINQ640R HELP disk address =

ENTER

19D is the default.

DMSINQ764R Language id =

langid

langid is the identifier for the 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

The system national language does not use a level id.

DMSINQ296R Should the installation segment be used? Enter 1 (YES) or 0 (NO).

The installation segment is an optional DCSS into which you can place frequently-used EXECs and System Product Editor 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 installation segment, or press **ENTER** to accept the default name, CMSINST.

```
DMSINI729R Do you want to save the system?  Enter 1 (YES) or 0 (NO).
```

no	The first time through this procedure you need to respond NO. When we put the new nucleus out on the 190 system disk, then you need to respond YES.

```
DMSINI730R Saved system name =
```

ENTER or ***systemname***	Since you responded NO to the previous prompt, you do not receive this prompt at this time. When you use the 190 minidisk to contain the new nucleus and respond YES to save the system, you receive this prompt and should specify CMS as the system name. 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 disk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

390	Note that when we put the new nucleus out on the 190 disk, you need to specify 190 at that time.

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn	***nnnnn*** is the cylinder/block location on 390 where the new nucleus will reside according to the device type and minidisk size you are using. Refer to the table on page 4-7. Use the value you used as the recomp value when formatting the 390.

```
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 disk 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, refer to the description of the IPL command in the *VM/SP CP 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 identification, **VM/SP Release** *n 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

The nucleus is then written on the specified disk cylinder and the version identification is displayed, indicating that the CMS system is loaded successfully and is ready to accept CMS commands.

VM/SP Release *n mm/dd/yy hh:mm:ss*	Or your own version identification.

You can use this copy of CMS to test updates and changes. Before you test the CMS system, create a backup disk file from the CMS nucleus and the nucleus load map. Issue the following commands:

spool reader nohold

close printer — This places the load map in your reader.

PRT FILE nnnn TO MAINT COPY 001 NOHOLD

close reader — This sends a copy of the CMS load map to your reader.

Now you can read a copy of the CMS nucleus and a copy of the CMS load map from your reader onto the 191 disk. Enter the following commands:

readcard cmsnuc nucleus a1 — Reads your nucleus to disk.

spgen map cms — Reads in the load map.

After you test the new CMS nucleus on 390 (you can do this by issuing some CMS commands), and you are satisfied that it is working correctly, you can use the disk file to create the new nucleus on the system disk (190).

To regenerate a nucleus which exists as a disk file (CMSNUC NUCLEUS for example), issue the following commands:

spool punch to *

punch cmsnuc nucleus a1 (noheader)

ipl 00c — Here you are IPLing your reader which contains the CMSNUC NUCLEUS file.

You may then answer the IPL messages previously described. This time:

Specify 190 as the system disk address instead of 390.

Answer YES to save the system.

Specify CMS as the saved system name.

Specify 190 as the IPL address instead of 390, and enter the correct cylinder or FB-512 block for your system disk.

Then, you can save the CMS saved system, if you wish.

Note: If a named system has been built from this CMS system disk, it must be resaved because the SSTAT is recreated only when the disk is loaded (for example 190).

Appendix L. Control File Identifiers

IBM uses file identifiers as listed below for distributed updates to VM/SP. Use the one CNTRL file that depicts your system configuration.

DMKSP CNTRL is used for CP source, copy, and macro updates with support for the uniprocessor. Its contents are:

```
TEXT MACS DMKSP DMKMAC DMSSP CMSLIB OSMACRO
TEXT AUXSP
```

DMKSPA CNTRL is used for CP source, copy, and macro updates with support for the attached processor. It should be used for all updates if attached processor support is required. Its contents are:

```
TEXT MACS DMKSPA DMKSP DMKMAC DMSSP CMSLIB OSMACRO
AP   UPDTAP
TEXT AUXSP
```

DMKSPM CNTRL is used for CP source, copy, and macro updates with support for the multiprocessor system. It should be used for all updating if multiprocessor support is required. Its contents are:

```
TEXT MACS DMKSPM DMKSP DMKMAC DMSSP CMSLIB OSMACRO
MP   UPDTMP
AP   UPDTAP
TEXT AUXSP
```

DMKSPT CNTRL is used for CP source, copy, and macro updates with support for the uniprocessor systems. It should be used for all updates if CP FRET Trap support is desired. Its contents are:

```
TEXT MAC DMKSPT DMSSP DMSMAC DMSSP CMSLIB OSMACRO
FT UPDTFT
TEXT AUXSP
```

DMKSPAT CNTRL is used for CP source, copy, and macro updates with support for the attached processor. It should be used for all updates if attached processor support is required and CP FRET Trap support is desired. Its contents are:

```
TEXT MACS DMKSPAT DMKSP DMSMAC DMSSP CMSLIB OSMACRO
APFT UPDTFT
AP   UPDTAP
TEXT AUXSP
```

DMKSPMT CNTRL is used for CP source, copy, and macro updates with support for the multiprocessor system. It should be used for all updates if multiprocessor support is required and CP FRET Trap support is desired. Its contents are:

```
TEXT MACS DMKSPMT DMKSP DMSMAC DMSSP CMSLIB OSMACRO
MP   UPDTMP
APFT UPDTFT
AP   UPDTAP
TEXT AUXSP
```

DMMSP CNTRL is used for IPCS source, copy and macro updates. Its contents are:

```
TEXT MACS DMMSP DMKSP DMKMAC CMSLIB OSMACRO
TEXT AUXSP
```

DMSMSP CNTRL is used for CMS copy and macro updates. Its contents are:

```
TEXT MACS
TEXT AUXMSP
```

DMSSP CNTRL is used for CMS source updates. Its contents are:

```
TEXT MACS DMSSP CMSLIB OSMACRO DOSMACRO TSOMAC DMKSP
TEXT AUXSP
```

CSISP CNTRL is used for GCS source updates. Its contents are:

```
TEXT MACS CSISP DMSSP CMSLIB DMKSP
TEXT AUXSP
```

ATSSP CNTRL is used for TSAF source updates. Its contents are:

```
TEXT MACS DMSSP CMSLIB OSMACRO DMKSP
TEXT AUXSP
```

Appendix M. Regenerating CP/CMS/IPCS Modules/Segments

Use this appendix when applying updates and local fixes. When service is shipped via PUT, all modules are regenerated and replaced on a one-to-one basis. When CMS source code or text decks are updated, the CMS nucleus and/or some CMS modules/segments must be regenerated. If you must regenerate the CMS nucleus, see "Building the New CMS Nucleus" on page K-3. Figure M-1 on page M-4 shows which modules/segments must be regenerated in each case. CMS should be resaved whenever the system S-disk is updated. This will ensure that the S-disk directory that is part of the saved CMS nucleus correctly matches the physical S-disk directory. See "Building the New CMS Nucleus" on page K-3.

Creating CMS Disk-Resident Modules

The CMSGEND EXEC procedure creates CMS disk-resident command modules from CMS text files. (Refer to "CMSGEND EXEC" on page 15-3 for more information about the CMSGEND EXEC). CMSGEND is invoked by specifying the filename of the module to be generated. For example, if there is a change to the text file DMSCMP, you must generate a new COMPARE MODULE.

Access the CMS service minidisk where the updated DMSCMP text file exists as your A-disk. Issue:

```
access 293 a
```

Access the 193 disk as a read-only extension of your A-disk to get access to the CMSGEND EXEC. Issue:

```
access 193 f
```

Issue the CMSGEND EXEC to generate a new COMPARE module on your A-disk:

```
cmsgend compare
```

After an existing file of 'COMPARE MODOLD A1' is erased, CMSGEND renames any existing file from 'COMPARE MODULE A2' to 'COMPARE

MODOLD A1'. CMSGEND then loads the text files that comprise the COMPARE command module and generates a new COMPARE MODULE A2.

When you use CMSGEND, you must have all pertinent text files available. The text files must have a filetype of TEXT; thus, if you have updated an object module using VMFASM, and the most recent object file has a filetype such as TXTLOC1, you must rename it to a filetype of TEXT. (Note that if there is currently a text file on the system disk, you may want to rename it also, so that your updated text file, on some other disk, is loaded.)

The following is a sample of what CMSGEND might display when it executes:

```
*** CURRENT STATUS:
FILE ' COMPARE MODULE A2' DOES NOT EXIST
FILE ' COMPARE MODOLD A1' DOES NOT EXIST

*** LOADING:
 INVALID CARD - D21371DS 313 UV15798 BALR INTO SYSTEM CODE PSW ENABLED
 INVALID CARD - *    DMSCMP   D21371DS B1 SP3393 mm/dd/yy hh:mm
 INVALID CARD - *    DMSSP    MACLIB   A2 SP3293 mm/dd/yy hh:mm
 INVALID CARD - *    CMSLIB   MACLIB   A2 RnM190 mm/dd/yy hh:mm
 INVALID CARD - *    DOSMACRO MACLIB   A2 RnM190 mm/dd/yy hh:mm
 INVALID CARD - *    DMSACC   ASSEMBLE A1 RnM303 mm/dd/yy hh:mm
                .
                .
                .
 INVALID CARD - *    DMSCMP   ASSEMBLE A1 SP3193 mm/dd/yy hh:mm
 DMSCMP   SD 00E000
 COMPARE     00E000

*** RESULTS:

'COMPARE MODULE A2' CREATED FROM TEXT DECK ( S ) DMSCMP
 WITH OPTIONS TRANS SYSTEM NOMAP
Ready;
```

CMSGEND invokes LOAD, which responds with warning messages about comment cards in your textfile, and builds the MODULE. You now have a new COMPARE MODULE on your A-disk which incorporates the updated DMSCMP TEXT from the A-disk. The new COMPARE MODULE should now be tested before you move it to the S-disk where everyone can use it.

Loading CMS Disk-Resident Modules

Remember that the S- and Y-disk directories are saved in the CMS saved system, so you must resave CMS when you change the S- or Y-disks.

The following steps put your new COMPARE MODULE (from "Creating CMS Disk-Resident Modules," above) into production, starting with the new textfile and MODULE on your A-disk.

Get write access to the 190 minidisk by issuing:

link * 190 190 mr

access 190 g

To avoid fragmentation on the S- and Y-disks for performance reasons, it is good practice to remove obsolete files. However, do not erase the old textfile on the 190 minidisk since you might need it later.

Rename the old textfile on the 193 making it mode 5. As a mode 5 file, users are not aware of it. Thus, you can erase it when you are sure you no longer need it. To do this, issue:

rename dmscmp text f1 = oldtext f5

Since the old textfile was mode 1, copy over the new textfile as mode 1. by issuing the following command:

copyfile dmscmp text a = = f1 (olddate

Rename the old COMPARE MODULE F2 to COMPARE OLDMOD F5. By making it mode 5, users are not aware of it. Thus, you can erase it when you're sure you no longer need it. Rename the new one COMPARE MODULE F2 by issuing:

rename compare module f2 = oldmod f5

copy compare module a2 = = f2 (olddate

IPL 190 to rebuild the nucleus SSTAT. (If you IPL CMS, the SSTAT will not be rebuilt.) To rebuild the CMS saved system and incorporate this new SSTAT, immediately issue the IPL command with the SAVESYS parameter.

```
ipl 190 parm savesys cms
SYSTEM SAVED
VM/SP Release n  mm/dd/yy hh:mm:ss
```

Or your own version identification.

```
ENTER
DMSACC723I Y (19E) R/O
```

This message appears only if the Y-STAT is too large to fit into the CMS nucleus. If this happens, you can generate a larger CMS nucleus that allows space for the Y-STAT. See Appendix F, "Enlarging the CMS Nucleus To Contain the Y-Disk Directory (Y-STAT)" on page F-1 for additional information.

```
Ready;
```

Make sure you have the correct filemode when you move something to the S- or Y-disks. Remember: The mode of a new file is determined by the mode of the file it is going to replace.

Note: It is recommended that changes to the S- and Y-disks be made while no other users are logged on the system.

Regenerating CMS Modules/Segments from Updated CMS Source

If you apply any corrective service to a text deck listed in Figure M-1 on page M-5, you must regenerate the corresponding CMS module or segment. If a source name does not appear in Figure M-1 on page M-5, the file is contained within the CMS nucleus.

Note: Make sure that you have access to the necessary minidisks; see "Disk Addresses for Updating" on page 11-7.

The CMSGEND EXEC procedure used to regenerate CMS modules is described on page 15-3. If you must regenerate the CMSDOS, CMSBAM, CMSAMS, or CMSVSAM segments, see Chapter 6, "Installing Discontiguous Saved Segments" for information about the corresponding EXEC procedures.

Footnotes for Figure M-1 on page M-5 are explained on page M-7.

Change in Source	Requires Regeneration of Module/Segment	Using EXEC Procedure
DMSAMS	AMSERV	CMSGEND
DMSASD	ASSEMBLE	CMSGEND
DMSASM	ASSEMBLE	CMSGEND
DMSASN	ASSGN	CMSGEND
DMSBAB	CMSDOS	DOSGEN
DMSBCT	DMSDFT	CMSGEND
DMSBLG	DMSDFT	CMSGEND
DMSBOP	CMSDOS	DOSGEN
DMSBTB	CMSBATCH	CMSGEND
DMSBTP[1]		
DMSCCK	CATCHECK	CMSGEND
DMSCDI	DMSDFT	CMSGEND
DMSCLS	CMSDOS	DOSGEN
DMSCMP	COMPARE	CMSGEND
DMSCVH	CMSDOS	DOSGEN
DMSDAS	CMSDOS	DOSGEN
DMSDFT	DMSDFT	CMSGEND
DMSDLK	DOSLKED	CMSGEND
DMSDMP	CMSDOS	DOSGEN
DMSDOS	CMSDOS	DOSGEN
DMSDSK	DISK	CMSGEND
DMSDSL	DOSLIB	CMSGEND
DMSDSV	DSERV	CMSGEND
DMSEDC	EDIT[2]	CMSGEND
DMSEDF	EDIT[2]	CMSGEND
DMSEDI	EDIT[2]	CMSGEND
DMSEDX	EDIT[2]	CMSGEND
DMSEND	CMSAMS	VSAMGEN (AMS)
DMSETR	CMSDOS	DOSGEN
DMSEXG	DCSSGEN	CMSGEND
DMSEXM	EXECMAP	CMSGEND
DMSFCH	CMSDOS	DOSGEN
DMSFOR	FORMAT	CMSGEND
DMSGIO	EDIT[2]	CMSGEND
DMSGLB	GLOBAL	CMSGEND
DMSGMF	CMSDOS	DOSGEN
DMSGND	GENDIRT	CMSGEND
DMSGTM	CMSDOS	DOSGEN
DMSGVE	CMSDOS	DOSGEN
DMSHDI	HNDINT	CMSGEND
DMSHDS	HNDSVC	CMSGEND
DMSHLB	HELPCONV	CMSGEND
DMSHLD	HELPCONV	CMSGEND
DMSHLE	HELPCONV	CMSGEND
DMSHLI	HELPCONV	CMSGEND
DMSHLP	HELPCONV	CMSGEND
DMSHLS	HELPCONV	CMSGEND
DMSICP	IOCP	CMSGEND
DMSIMA	IMAGEMOD	CMSGEND
DMSLAB	CMSDOS	DOSGEN
DMSLBD	LABELDEF	CMSGEND

Figure M-1 (Part 1 of 3). Regenerating CMS Modules/Segments from Updated CMS Source

Change in Source	Requires Regeneration of Module/Segment	Using EXEC Procedure
DMSLBM	MACLIB	CMSGEND
DMSLBT	TXTLIB	CMSGEND
DMSLBR[3]		
DMSLCK	CMSDOS	DOSGEN
DMSLDF	CMSDOS	DOSGEN
DMSLDS	LISTDS	CMSGEND
DMSLIC	CMSDOS	DOSGEN
DMSLKD[3]		
DMSLLU	LISTIO	CMSGEND
DMSLMX	TAPE[4]	CMSGEND
DMSMCM	CMSDOS	DOSGEN
DMSMDP	MODMAP	CMSGEND
DMSMGC	GENMSG	CMSGEND
DMSMGD	GENMSG	CMSGEND
DMSMGE	GENMSG	CMSGEND
DMSMVE	MOVEFILE	CMSGEND
DMSMVG[5]		
DMSNXD	NUCXDROP	CMSGEND
DMSNXM	NUCXMAP	CMSGEND
DMSOPL	CMSDOS	DOSGEN
DMSOPT	OPTION	CMSGEND
DMSOR1	CMSDOS	DOSGEN
DMSOR2	CMSDOS	DOSGEN
DMSOR3	CMSDOS	DOSGEN
DMSOSR	OSRUN	CMSGEND
DMSOVR	SVCTRACE	CMSGEND
DMSOVS	DMSOVS	CMSGEND
DMSPCA	DMSPCC	CMSGEND
DMSPCB	DMSPCC	CMSGEND
DMSPCC	DMSPCC	CMSGEND
DMSPCR	DMSPCC	CMSGEND
DMSPCT	DMSPCC	CMSGEND
DMSPCW	DMSPCC	CMSGEND
DMSPOA	PROP	CMSGEND
DMSPOC	PROP	CMSGEND
DMSPOD	PROP	CMSGEND
DMSPOE	PROP	CMSGEND
DMSPOL	PROP	CMSGEND
DMSPON	PROP	CMSGEND
DMSPOP	PROP	CMSGEND
DMSPOQ	PROP	CMSGEND
DMSPOR	PROP	CMSGEND
DMSPOS	PROP	CMSGEND
DMSPRE	PRELOAD	CMSGEND
DMSPRV	PSERV	CMSGEND
DMSPUN	PUNCH	CMSGEND
DMSRDC	READCARD	CMSGEND
DMSRDR	RDR	CMSGEND
DMSRNE	RENUM	CMSGEND
DMSRPG	CMSDOS	DOSGEN

Figure M-1 (Part 2 of 3). Regenerating CMS Modules/Segments from Updated CMS Source

Change in Source	Requires Regeneration of Module/Segment	Using EXEC Procedure
DMSRRV	RSERV	CMSGEND
DMSRSV	RESERVE	CMSGEND
DMSSCR	EDIT[2]	CMSGEND
DMSSFD	DMSSFD	CMSGEND
DMSSMG	DMSDFT	CMSGEND
DMSSPR	SETPRT	CMSGEND
DMSSRT	SORT	CMSGEND
DMSSRV	SSERV	CMSGEND
DMSSSK	SETKEY	CMSGEND
DMSSTX	CMSDOS	DOSGEN
DMSSUB	CMSDOS	DOSGEN
DMSSVL	CMSDOS	DOSGEN
DMSSYN	SYNONYM	CMSGEND
DMSTMA	TAPEMAC	CMSGEND
DMSTPD	TAPPDS	CMSGEND
DMSTPE	TAPE[4]	CMSGEND
DMSTPF	TAPE[4]	CMSGEND
DMSTPG	TAPE[4]	CMSGEND
DMSTPH	TAPE[4]	CMSGEND
DMSTPI	TAPE[4]	CMSGEND
DMSTPJ	TAPE[4]	CMSGEND
DMSTYP	TYPE	CMSGEND
DMSUPD	UPDATE	CMSGEND
DMSUSR	DMSDFT	CMSGEND
DMSUTL	LOADLIB	CMSGEND
DMSVAN	CMSAMS	VSAMGEN (AMS)
DMSVAS	CMSAMS	VSAMGEN (AMS)
DMSVAX	CMSAMS	VSAMGEN (AMS)
DMSVBM[6]		
DMSVIP	CMSVSAM	VSAMGEN (VSAM)
DMSVIS	CMSDOS	DOSGEN
DMSVLT	CMSDOS	DOSGEN
DMSVVN	CMSVSAM	VSAMGEN (VSAM)
DMSVVS	CMSVSAM	VSAMGEN (VSAM)
DMSXCP	CMSDOS	DOSGEN
DMSXMS	DMSXMS[7]	CMSGEND
DMSXRE	DMSXRE[7]	CMSGEND
DMSZAP	ZAP	CMSGEND
DMSZIT	EDIT[2]	CMSGEND

Figure M-1 (Part 3 of 3). Regenerating CMS Modules/Segments from Updated CMS Source

Notes:

1 DMSBTP is loaded into free storage by DMSBTB. The file DMSBTP TEXT must be filemode 2 on the system disk.

2 When the CMSGEND EXEC procedure is invoked for EDIT, it creates the EDIT module. Then CMSGEND automatically reinvokes itself to create the EDMAIN module.

[3] Update procedures are not provided for DMSLBR and DMSLKD. DMSLBR is shipped as part of CMSBAM DOSLIB. DMSLKD is contained in the LKED module.

[4] When the CMSGEND EXEC procedure is invoked for TAPE, it creates the TAPE module from DMSLMX. Then CMSGEND automatically reinvokes itself to create the DMSTPI module from the remaining text files.

[5] DMSMVG is loaded into free storage by DMSMVE. The file DMSMVG TEXT must be filemode 2 on the system disk.

[6] Update procedures are not provided for DMSVBM. DMSVBM is shipped as part of the CMSBAM DOSLIB.

[7] All EDIT source files, except DMSXMS and DMSXRE, are contained within the CMS nucleus.

Regenerating System Product Interpreter Programs from Updated Source

If you apply a PTF to a System Product Interpreter source program, use the EXECUPDT command to create an executable version of the program. You can use this command ONLY with System Product Interpreter programs. Figure M-2 indicates the options that you should select in each case.

If the source program is an EXEC or System Product Editor macro that is included in the installation DCSS (default name CMSINST), you should use the NOCOMMENTS option with the EXECUPDT command to remove all comments and leading blanks. Then you must reload and save the installation DCSS to include the updated program (see "Installing CMSINST" on page 6-16).

In the list of options for the EXECUPDT command, "fn" is an UPDATE control file that controls the application of multiple UPDATE files to the source input file. For more information about the EXECUPDT command, refer to the *VM/SP CMS Command Reference*.

Source		EXEC Procedure	Options
ALL	$XEDIT	EXECUPDT	CTL fn HIST SID
CANCEL	$XEDIT	EXECUPDT	CTL fn HIST SID
CAPPEND	$XEDIT	EXECUPDT	CTL fn HIST SID
CMSSERV	$EXEC	EXECUPDT	CTL fn HIST SID
DEFAULTS	$EXEC	EXECUPDT	CTL fn HIST SID
DIRECGEN	$EXEC	EXECUPDT	CTL fn HIST SID
DISCARD	$EXEC	EXECUPDT	CTL fn HIST SID
DISKMAP	$EXEC	EXECUPDT	CTL fn HIST SID
EXECUPDT	$EXEC	EXECUPDT	CTL fn HIST SID
EXECUTE	$XEDIT	EXECUPDT	CTL fn HIST SID
EXPAND	$EXEC	EXECUPDT	CTL fn HIST SID

Figure M-2 (Part 1 of 2). Regenerating System Product Interpreter Programs from Updated Source

Source		EXEC Procedure	Options
GCSGEN	$EXEC	EXECUPDT	CTL fn HIST SID
HELP	$XEDIT	EXECUPDT	CTL fn HIST SID
HELPXED	$XEDIT	EXECUPDT	CTL fn HIST SID
JOIN	$XEDIT	EXECUPDT	CTL fn HIST SID
LANGGEN	$EXEC	EXECUPDT	CTL fn HIST SID
LANGMERG	$EXEC	EXECUPDT	CTL fn HIST SID
MACLIST	$EXEC	EXECUPDT	CTL fn HIST SID
MOREHELP	$EXEC	EXECUPDT	CTL fn HIST SID
MSGFAC	$EXEC	EXECUPDT	CTL fn HIST SID
MSGRESP	$EXEC	EXECUPDT	CTL fn HIST SID
PARSE	$XEDIT	EXECUPDT	CTL fn HIST SID
PREFIXX	$XEDIT	EXECUPDT	CTL fn HIST SID
PRFSHIFT	$XEDIT	EXECUPDT	CTL fn HIST SID
PRFSHOW	$XEDIT	EXECUPDT	CTL fn HIST SID
PROFMLST	$XEDIT	EXECUPDT	CTL fn HIST SID
PROPEPIF	$EXEC	EXECUPDT	CTL fn HIST
PROPHCHK	$EXEC	EXECUPDT	CTL fn HIST
PROPLGER	$EXEC	EXECUPDT	CTL fn HIST
PROPPCHK	$EXEC	EXECUPDT	CTL fn HIST
PROPPROF	$EXEC	EXECUPDT	CTL fn HIST
PROPRTCV	$EXEC	EXECUPDT	CTL fn HIST
PROPST	$EXEC	EXECUPDT	CTL fn HIST
RGTLEFT	$XEDIT	EXECUPDT	CTL fn HIST SID
SI	$XEDIT	EXECUPDT	CTL fn HIST SID
SNTINFO	$EXEC	EXECUPDT	CTL fn HIST SID
SNTMAP	$EXEC	EXECUPDT	CTL fn HIST SID
SPLTJOIN	$XEDIT	EXECUPDT	CTL fn HIST SID
STATUS	$XEDIT	EXECUPDT	CTL fn HIST SID
SYSPROF	$EXEC	EXECUPDT	CTL fn HIST SID
VMFLKED	$EXEC	EXECUPDT	CTL fn HIST SID
X$EUPD$X	$XEDIT	EXECUPDT	CTL fn HIST SID
X$MLST$X	$XEDIT	EXECUPDT	CTL fn HIST SID
ZAPTEXT	$EXEC	EXECUPDT	CTL fn HIST SID
$DASD$	$CONSTS	EXECUPDT	CTL fn SID

Figure M-2 (Part 2 of 2). Regenerating System Product Interpreter Programs from Updated Source

Regenerating CMS Modules from Updated Utility Source Programs

If you apply PTFs to certain utility source programs, the corresponding CMS modules must also be regenerated to run properly. Figure M-3 shows the source name, module name, and the EXEC procedure that you use to regenerate the module.

Note: Make sure that you have access to the necessary minidisks; see "Disk Addresses for Updating" on page 11-7.

CMSGEND EXEC is described on page 15-3, UTILITY EXEC on page 15-36, and INSTEP EXEC on page J-4.

Change in Source	Requires Regeneration of Module	EXEC Procedure To Use
CSIYTD	TRAPRED	CMSGEND
CSIYTS	TRAPRED	CMSGEND
DMKDDC	DDR	UTILITY or CMSGEND
DMKDDR	DDR	UTILITY or CMSGEND
DMKDDT	DDR	UTILITY or CMSGEND
DMKDIR	DIRECT	UTILITY or CMSGEND
DMKDNC	DDR	UTILITY or CMSGEND
DMKDNT	DDR	UTILITY or CMSGEND
DMKFMT	None, no CMS execution	UTILITY
DMKIMG	GENIMAGE	CMSGEND
DMKNMT	IMAGELIB	CMSGEND
DMKOVR	OVERRIDE	CMSGEND
DMKRND	NCPDUMP	CMSGEND
DMKTEE	TRAPRED	CMSGEND
DMKTEF	TRAPRED	CMSGEND
DMKTEM	TRAPRED	CMSGEND
DMKTES	TRAPRED	CMSGEND
DMKTRR	TRAPRED	CMSGEND
DMSARD	ASM3705	INSTEP
DMSARN	ASM3705	INSTEP
DMSARX	ASM3705	INSTEP
DMSGRN	GEN3705	CMSGEND
DMSNCP	SAVENCP	CMSGEND
VMFDATE	VMFDATE	CMSGEND
VMFDOS	VMFDOS	CMSGEND
VMFLOAD	VMFLOAD	CMSGEND
VRSIZE	VRSIZE	CMSGEND

Figure M-3. Regenerating CMS Modules from Updated Utility Source Programs

Modules to Regenerate from IPCS Text Files

If you update an IPCS text file shown the left-hand column of Figure M-4 on page M-11, you should use the GENMOD command to regenerate the module(s) specified on the right. For more information about the GENMOD command, refer to the *VM/SP CMS Command Reference*.

IPCS FILES	CLRSCRN	CONVERT	DUMPSCAN	IPCSDUMP	MAP	PROB	PRTDUMP	STAT	SUMMARY
DMMAMP DMMCHN DMMCLR DMMCMS	—		— — —	— —		—	—		
DMMCOM DMMCOR DMMCPA DMMCVD		— —	—		—		—		
DMMCVS DMMDCC DMMDCD DMMDCM									
DMMDCO DMMDCP DMMDID DMMDIR			— —						
DMMDSC DMMEDM DMMEDX DMMEXT			—						
DMMFED DMMFEX DMMHEX DMMINI			— — —	—					
DMMINT DMMIOB DMMLOC DMMMAP			— —		—				
DMMMOD DMMNMP DMMNUC DMMPRG			—						
DMMPRM DMMPRO DMMPRT DMMREG			— —	— —		—	—		
DMMSCR DMMSEA DMMSEG DMMSTA			— —	—		—		—	
DMMSUM DMMTAB DMMTIU DMMTRC			—	— — —		—	—		—
DMMVAL DMMVMB DMMVMF DMMWRT			—	— — —		—			

Figure M-4. Modules to Regenerate from IPCS Text Files

Appendix N. Servicing Systems Network Architecture (SNA) Products

This appendix contains information about how to service the Systems Network Architecture (SNA) products. These products are object maintained and service by replacing TEXT files. **Do not use this appendix to service any of the base components of VM/SP**. The execs we discuss in this appendix are:

VMFMERGE	applies PTFs form the delta disk to the merge disk.
VMFREMOV	removes PTFs applied by the VMFMERGE EXEC procedure
VMFZAP	applies ZAPs and maintains a record of them in the ZAP log.

All these execs are discussed in Chapter 15, "EXEC and Command Procedures" on page 15-1.

Before you try to do any service processing using these execs, you must consider the following to determine the amount of virtual storage that you need to define for your virtual machine.

- the number of minidisks accessed and the number of files on each minidisk accessed.
- the size of files (Merge Log, Reqby Log, Zap Log) needing to be updated, and other files used during processing.
- the number of EXECs loaded into virtual storage. You can find this out by using the CMS command EXECMAP. Refer to the *VM/SP CMS Command Reference* for more information about this command.
- the number of nucleus extensions loaded into virtual storage. You can find this out by using the CMS command NUCXMAP. Refer to the *VM/SP CMS Command Reference* for more information about this command.

A description of the types of disks, the files, logs and lists used during service processing follows. In addition, there are generic descriptions of how to do different types of object code service.

Types of Disks

VMFMERGE, VMFREMOV, and VMFZAP use several types of disks during processing. Although we refer to each of these as a single disk, there may be multiple occurrences of each type. The EXECs access these disks using mode letters E-N. When processing ends, the EXECs restore your disk hierarchy to its original condition.

The five types of disks are:

Disk	Contents
Base	Original product files as shipped on the product tape.
Delta	Changed portions of the product. These files have different names from the corresponding files that are found on the base disk. There is also a Service Control File (containing all the information needed to install a program temporary fix (PTF)) for each change on this disk. There is an Exclude List, at least one Apply List, and possibly a Remove List.
Merge	Changes that have been applied. These files have been copied from the delta disk and renamed to match the corresponding file on the base disk. There is also a log containing changes (Merge Log) which have been merged into the product and a log containing the requisite relationships (Reqby Log) of the merged or superseded changes.
ZAP	TEXT files that have been copied from the base or merge disk and ZAPped. There is a log containing ZAPs (ZAP Log) that have been applied to the product.
Run	Actual working version of the product. The files on this disk are created by the product build EXEC from the files on the base, merge, and ZAP disks. There is also a log containing service (Service Log) of PTFs and ZAPs that have been applied to the product.

Service Control File

The Service Control File (SCF) describes a Program Temporary Fix (PTF). There is a SCF for each PTF. SCFs are built by the change team and shipped with the PTFs in the delta file of the Program Update Tape (PUT). The filename of the SCF is the PTF number and the filetype must be SCF.

The Service Control File contains all the information needed to install a PTF. The following is an example of a Service Control File.

```
:ptf.UV00006
:prodid.5664175
:prereq.UV00005 UV00007
:coreq.UV00056.5748RC2
:sup.Z00002 Z00003
:changes.
 :element.DSIMNT TEXT
  :replace.TXTP0006
 :element.DSIXXX TEXT
  :replace.TXTP0004
 :element.DSIYYY TEXT
  :replace.TXTP0001
:echanges.
:apartext.PP00004 - Split DSIXXX for new base register
:apartext.PP00009 - Update loadlist to add DSINEW
```

Figure N-1. Example of a Service Control File (UV00006 SCF)

where

:ptf. is the PTF number. This number is the same as the filename of the SCF.

:prodid. is the seven character product identifier. (You may also specify a one character suffix for the release or level of the product. This suffix is determined by the service group).

:prereq. is the SCF filename(s) of any PTF(s) that must be merged before you can merge the PTF specified on the *:ptf.* tag. During processing, if the prerequisite cannot be found, then processing ends.

If a prerequisite change is not for this product, the associated *prodid* must be specified as part of the PTF name on the *:prereq.* tag. For example, *ppppppp.prodid*, where ppppppp is the PTF number within another product. A message is displayed by VMFMERGE telling you there is a prerequisite PTF for another product, specifically the product associated with the *prodid* listed on the *:prereq.* tag.

You can omit PRODID if the products are the same as for this PTF.

:coreq. is the SCF filename(s) of any PTF(s) that need to be merged together with this PTF.

If a corequisite change is not for this product, the same rules apply as with *:prereq* for specifying *prodid.* A message is displayed by VMFMERGE and VMFREMOV telling you there is a corequisite PTF for another product, specifically the product

associated with the *prodid* listed on the *:coreq.* tag. As shown in the example above, the corequisite UV00056 is for product 5748RC2.

The maximum number of characters (including blanks) that you can specify on a *:coreq.* tag is 256.

:sup. are the PTFs or ZAPs that are no longer needed as a result of this PTF. Specifying this tag prevents superseded PTFs or ZAPs from being reapplied.

:changes. indicates the beginning of the list of elements changed. The change list is a table with an entry for each element affected. Each element begins with an *:element* tag and ends with the next *:element* tag.

:element. specifies the CMS filename and filetype of an element defined or replaced by this PTF. A separate tag is coded for every element changed.

:replace. specifies the CMS filetype of an object replacement file for the associated element.

:echanges. indicates the ending of the list of elements changed.

:apartext. is an APAR number followed by description of the problem reported by the APAR. The service group specifies this information.

Product Parameter File

The Product Parameter file contains records that identify the various product minidisk addresses used for installation and service. A parameter file containing the default minidisk addresses is shipped with the product on the product tape. (You can change or add to the default addresses). The VMFMERGE, VMFREMOV, VMFZAP, and product EXECs read this file for the information needed to access minidisks. Before you use these EXECs, you must access the minidisk containing the product parameter file.

The filename of the file is the product identifier and the filetype must be VMFPARM.

Each record in the file contains a keyword indicating the type of disk and one or more virtual addresses. The format of the records is a keyword followed by one or more values. VMFMERGE, VMFREMOV, and VMFZAP recognize the following keywords.

Base the virtual address(es) of the minidisk(s) containing the base product code.

Delta the virtual address(es) of the Delta Disk(s).

Merge the virtual address(es) of the Merge Disk(s).

Zap the virtual address(es) of the Zap Disk(s).

Run the disk for executable code.

***** this record is treated as a comment.

Note: VMFMERGE, VMFREMOV, and VMFZAP ignore any records beginning with an unrecognized keyword. This allows products to define other keywords that product EXECs can use.

The following is an example of a product parameter file.

```
Base 250
Delta 251
Merge 252
ZAP 253
Run 254
```

Figure N-2. Example of a Product Parameter File

Note: You can specify more than one address on these disks. This may be valuable if you want to keep a record of the disks which have different levels of your system which you created from different Merge disks or Delta disks. The maximum number of disks you can access is ten.

During service processing, the EXECs read the product parameter file and access the disks that are needed. For example, VMFMERGE and VMFREMOV access the delta and merge disks; whereas, VMFZAP accesses the base, merge, and ZAP disks.

How VMFMERGE, VMFREMOV, and VMFZAP Use the Product Parameter File

- VMFMERGE, VMFREMOV, and VMFZAP get the minidisk information from the Parameter file and access the needed disks using filemodes E-N. Once these EXECs stop processing, the disk hierarchy is restored.

- VMFMERGE and VMFREMOV only access the first virtual address following the **Merge** keyword. However, these EXECs access multiple Delta disks in the order in which they are listed in the Parameter file. Input files are taken from the first minidisk where they are found; output files are directed to the first single accessed Merge disk.

- VMFZAP only accesses the first virtual address following the **Zap** keyword. However, this EXEC accesses multiple Merge and Base disks in the order in which they are listed in the Parameter file (the Merge disks come before the Base disks). Input files are taken from the first minidisk where they are found; output files are directed to the first single accessed ZAP disk.

  In order to use VMFZAP, you MUST have an A disk accessed Read-Write. This disk MUST NOT be the ZAP disk, Merge disk, or Base disk. That is, the virtual address of your A disk MUST NOT appear on the ZAP, Merge, or Base records of your VMFPARM file.

Merge Log

A Merge Log is a file maintained by the VMFMERGE and VMFREMOV EXECs. This file is a log of changes (i.e. PTFs) which have been merged and/or superseded. A Merge Log exists for each product and is shipped on all Program Update Tapes (PUTs) and on the product tape. Note that the filename of the log is the product identifier and the filetype must be VMFMGLOG.

VMFMERGE and VMFREMOV update the Merge Log on the disk where it currently exists. (The Merge Log must be on the merge disk.) VMFMERGE and VMFREMOV stop processing if they cannot find a Merge Log.

The Merge Log has an entry for each PTF indicating the PTF number and its status. Status can be:

Merged means the change is included in the code.

Superseded means the change is no longer needed since some later fix has replaced it.

In addition, each entry has information on the date and time that the change was processed, and a list of the elements (TEXTs, EXECs, etc.) the change affects.

Every time a change is processed, a new entry for that change is added to the Merge Log. This is referred to as the 'history' of the change. The Merge Log is a record of the history of PTFs applied to and removed from a product. To find out the current status of a change, you need to read your Merge Log file starting at the bottom. Find the PTF in question. The first non-commented entry for that PTF indicates the status of the change.

For VMFREMOV, the merged entry is commented out and another entry is added to the end of the Merge Log indicating that the PTF has been removed.

Note: If you change the Merge Log in any way, either of the following could happen:

- a change may not be merged
- a change may be merged but not removed

An example of a Merge Log produced by VMFMERGE[1] and updated by VMFREMOV follows.

---

1 The Merge Log loaded from the PUT tape has a different time and date stamp.

```
:entry.UV00001  Merged      12/13/83 13:09:42 DSILGN TEXT DSISRP TEXT
:entry.UV00005  Merged      12/15/83 06:28:17 DSIXSD TEXT DSIZVSIN TEXT
                           DSIREP TEXT DSIMCB TEXT DSITVB COPY
                           DSIMLG TEXT

:entry.UV00008  Merged      12/15/83 06:28:17 DSILMODE EXEC
*entry.UV00007  Merged      12/24/83 11:28:17 DSILMODE EXEC
:entry.UV00006  Merged      01/12/84 17:56:39 DSIXXX TEXT DSIMNT TEXT
:entry.Z00002   Superseded  01/12/84 17:56:39 By UV00006
:entry.Z00003   Superseded  01/12/84 17:56:39 By UV00006
*entry.UV00007  Removed      2/24/84 11:28:17 DSILMODE EXEC
```

Figure N-3. Example of a Merge Log Produced by VMFMERGE

Notes:

1. *Comment records may be included in the Merge Log file.*
2. *Each comment record must begin with an asterisk (*) in column 1.*
3. *You cannot put a comment record in the middle of an entry which spans more than one line in the file. (You should put all comment records at the beginning of the Merge Log file).*
4. *The VMFREMOV command may also insert comment records in the Merge Log.*
5. *Blank lines are allowed between entries in the file, and are ignored.*

ZAP Log

The ZAP Log is a file maintained by the VMFZAP EXEC. This file is a log of applied ZAPs. A ZAP Log exists for each product and is built and maintained by the VMFZAP EXEC. Each entry has information on the date and time that the change was processed, and a list of the elements (TEXT files) that the change affects. Note that the filename of the log is the product identifier and the filetype must be VMFZPLOG.

The ZAP Log contains information about TEXT files currently affected by ZAPs applied to a given product. This information includes the filename(s) of the ZAP control files(s), the time and date the ZAP was applied, and the modules affected by the ZAP. To remove ZAPs which may be superseded by service you applied, VMFZAP first uses this list of affected modules to erase the files which were previously ZAPped. The currently wanted ZAPs, which are not superseded by service you applied, are then applied and VMFZAP creates a new version of the ZAP Log. VMFZAP erases the old ZAP before it writes the new one.

The ZAP Log is on the ZAP disk. New ZAP Logs are always created on the ZAP disk. You must not move the ZAP Log from the ZAP disk.

The ZAP Log has an entry for each ZAP indicating the ZAP number and its status. Status can be:

Zapped means the change is included in the code.

An example of a ZAP Log follows.

```
* ZAP Log for Product 5664175
:entry.Z00001   Zapped   12/31/83 12:34:56 DSINMME TEXT
:entry.Z00004   Zapped   12/31/83 12:34:56 DSIMNT TEXT
:entry.Z00005   Zapped   12/31/83 12:34:56 DSIMNT TEXT
```

Figure N-4. Example of a ZAP Log (5664175 VMFZPLOG)

Note: Comment records may be included in the ZAP Log. Such records contain an asterisk (*) in column 1. Also, blank lines are allowed anywhere in the file and are ignored.

Reqby Log

A Reqby Log is a file maintained by the VMFMERGE and VMFREMOV EXECs. The Reqby Log resides on the Merge disk. This file is a log of all dependent PTFs of each PTF which has a merged or superseded entry in the Merge Log. A dependent PTF is one which either has a given PTF as a prerequisite or corequisite. A Reqby Log exists for each product and is shipped on all Program Updates Tapes (PUTs) and on the product tape. Note that the filename of the log is the product identifier and the filetype must be VMFREQBY.

The Reqby Log is not required when you issue VMFMERGE or VMFREMOV. If the log does not exist, VMFMERGE or VMFREMOV automatically creates one using the information in the existing Merge Log and all Service Control Files.

The Reqby Log contains two types of entries: a comment entry and dependent entries.

A comment entry contains an asterisk (*) in column 1. Only the first non-blank line in the Reqby Log should be a comment line. VMFMERGE and VMFREMOV ignore all other comments and eliminate the comments whenever the Reqby Log is processed. If the first non-blank line is not a comment, one is automatically created the next time VMFMERGE or VMFREMOV processes the Reqby Log.

The dependent entries are identified by an *:entry.* tag followed by a particular PTF and its dependents. Dependent entries can overflow to the next line (or lines) whenever there are more dependent changes associated with the PTF than fit on one line. You can have as many overflow lines as needed to list all the dependent changes. Blank lines are allowed anywhere in the file and are ignored.

Note: If you change the Reqby Log in any way, either of the following could happen:

- a change may not be merged
- a change may be merged but not removed

An example of a Reqby Log follows.

```
* 5664167 VMFREQBY
:entry.ptf1 pft2 ptf3
:entry.ptf2 ptf1
```

Figure N-5. Example of a Reqby Log (5664167 VMFREQBY)

In the above example, PTF1 and PTF2 are corequisites and PTF1 is a prequisite of PTF3.

Service Log

The Service Log is the Merge Log with the ZAP Log appended on the end. It contains information about service that has been applied to your product. A Service Log exists for each product and is built by the product build EXECs.[2] The build EXEC copies the Merge Log for the product and appends the ZAP Log to it. Note that the filename of the Service Log is the product identifier and the filetype must be VMFSVLOG.

The format of the Service Log is a combination of the formats of the Merge Log (see Figure N-3 on page N-8) and the ZAP Log (see Figure N-4 on page N-9). By viewing the Service Log, you can determine whether a PTF or ZAP has been applied to your product.

An example of a Service Log follows.

```
:entry.UV00001  Merged      12/13/83 13:09:42 DSILGN TEXT DSISRP TEXT
:entry.UV00005  Merged      12/15/83 06:28:17 DSIXSD TEXT DSIZVSIN TEXT
                           DSIREP TEXT DSIMCB TEXT DSITVB COPY
                           DSIMLG TEXT

*:entry.UV00007  Merged      12/15/83 06:28:17 DSILMODE EXEC
:entry.UV00006  Merged      01/12/84 17:56:39 DSIXXX TEXT DSIMNT TEXT
:entry.Z00002   Superseded  01/12/84 17:56:39
:entry.Z00003   Superseded  01/12/84 17:56:39
*:entry.UV00007  Removed      2/25/84 06:28:17 DSILMODE EXEC
* ZAP Log for Product 5664175 created by MAINT 12/31/83  12:34:56
:entry.Z00001   Zapped      12/31/83 12:34:56 DSINMME TEXT
:entry.Z00004   Zapped      12/31/83 12:34:56 DSIMNT TEXT
:entry.Z00005   Zapped      12/31/83 12:34:56 DSIMNT TEXT
```

Figure N-6. Example of a Service Log (5664175 VMFSVLOG)

Note: Comment records may be included in the Service Log. Such records contain an asterisk (*) in column 1. Also, blank lines are allowed anywhere in the file and are ignored.

---

[2] The Service Log resides on the 'run-time' disk.

Apply List

The Apply List is a file which lists PTFs to be applied to a product. There is an Apply List shipped on a Program Update Tape (PUT). That file contains the names of all PTFs in the Delta file of the PUT. (The filename of that Apply List is *prodid*). Apply Lists must be on a delta disk.

You can build and then maintain the file yourself. There is an Apply List for each product which has PTFs applied to it. If no PTFs are applied to a product, you do not need to create an Apply List.

Any Apply Lists you maintain should not have the prodid as the filename. Each time you load the Delta files from the tape, the *prodid* Apply List on the Delta disk will be replaced with the *prodid* Apply List from the tape. If you want to change the supplied Apply List, make a copy of the supplied Apply List (give it a different filename) and change the 'copied' version of the Apply List.

The first word on each line in the file is the filename of a PTF. Only one PTF name should be placed on a line, the remaining data on a line is treated as a comment. If the first character on the line is an asterisk (*), then the whole line is treated as a comment. In addition, blank lines are allowed anywhere in the file and are ignored.

An example of an Apply List follows. Note that you can specify the filename[3] -- in this example the prodid is used -- but the filetype must be APPLIST.

```
* Apply List for Product 5664175
UV00006  PTF for maintenance program
UV00003
*UV00009
```

Figure N-7. Example of an Apply List (5664175 APPLIST)

[3] **The exception is that you cannot have a filename of EXCLUDE.**

Remove List

The Remove List is a file which lists the PTFs you want to remove from a product. You must build and maintain the file yourself. Remove Lists must be on a delta disk.

The first word on each line in the file is the filename of a PTF. Only one PTF name can be placed on a line. The remaining data on a line is treated as a comment. If the first character on the line is an asterisk (*), then the whole line is treated as a comment. Blank lines are allowed anywhere in the file and are ignored.

An example of a Remove List follows. Note that you can specify the filename -- in this example the prodid is used -- but the filetype must be REMLIST.

```
* Remove List for Product 5664175
UV00001  this is a comment for ptf1
UV00004

*UV00005  this is a comment for ptf5
```

Figure N-8. Example of a Remove List (5664175 REMLIST)

Exclude List

An Exclude List is a file listing PTFs to be excluded from a product.

There is an Exclude List shipped as part of the Delta File on the Program Update Tape (PUT). This list contains the names of PTFs known to be in error. (The filename of that Exclude List is *prodid*). The VMFMERGE EXEC uses the combination of the Exclude List you have and the *prodid* EXCLIST to obtain the names of the PTFs to exclude from the product. Exclude lists must be on a delta disk.

You can build and maintain the file yourself. Any Exclude Lists you maintain should not have the prodid as the filename. Each time you load the Delta files from the tape, the *prodid* Exclude List on the Delta disk will be replaced with the *prodid* Exclude List from the tape. If you want to change the supplied Exclude List, make a copy of the supplied Exclude List (give it a different filename) and change the 'copied' version of the Exclude List.

The first word on each line in the file is the filename of a PTF. Only one PTF name should be placed on a line, the remaining data on a line is treated as a comment. If the first character on the line is an asterisk (*), then the whole line is treated as a comment. In addition, blank lines are allowed anywhere in the file and are ignored.

An example of an Exclude List follows. Note that you can specify the filename -- in this example the prodid is used -- but the filetype must be EXCLIST.

```
* Exclude List for Product 5664175
UV00002 Exclude PTF from CP ACCESS command.
UV00013
*UV00016
```

Figure N-9. Example of an Exclude List (5664175 EXCLIST)

ZAP List

A Zap List is a file listing ZAPs to be applied to a product. You must build and maintain the file yourself.

The VMFZAP EXEC uses this file during its processing. If no ZAPs are applied then you do not need to create a ZAP List. ZAP lists must be on a disk listed on the BASE, MERGE, or ZAP entry record of the VMFPARM file for the product.

The first word on each line in the file is the filename of a ZAP. Only one ZAP name should be placed on a line, the remaining data on a line is treated as a comment. If the first character on the line is an asterisk (*), then the whole line is treated as a comment. In addition, blank lines are allowed anywhere in the file and are ignored.

If you want to 'back off' an unwanted ZAP, comment that ZAP out in the ZAP List by placing an asterisk (*) in the first column of the line containing the ZAP filename you want to 'back off'. Then run VMFZAP.

An example of a ZAP List follows. Note that the filename must be the **prodid** and the filetype must be ZAPLIST.

```
* Zap List for Product 5664175
Z00001 Zap for CP ACCESS command.
Z00004
Z00005
*Z00016
```

Figure N-10. Example of a Zap List (5664175 ZAPLIST)

Object Code Service Processing

There are many ways to apply service to components of your system that are object code maintained. This section describes, in a very generic way, how to:

- Apply
  - Emergency fixes (using ZAPs)
  - Corrective Service
  - Preventive Service
- Merge Service
- Remove Service
- Prevent regression
- Remove a fix-in-error

Apply Emergency Fixes Using ZAPs

When you need an emergency fix, you normally call in the problem and receive a fix in the form of a ZAP, over the phone.

Suppose there is a problem with module A. Here's what you need to do.

1. First call the IBM Support Center and report the problem.

2. You will be given a ZAP (i.e. Z00005) to apply to module A.

3. You must create a ZAP control file (Z00005 ZAP).

   For more information about valid records in the ZAP control file, refer to "Input Control Records" on page 15-99.

4. You must create a ZAP list if you do not already have one for the product. If you already have one, you must append this new entry.

   Creating a ZAP List - create a new file (on a disk accessed in the VMFPARM file) called *prodid* ZAPLIST. Enter the ZAP name (Z00005) as the first word on the line. (The rest of the information on each line is treated as a comment.)

   Adding to an existing ZAP List - on a new line at the bottom of the file, add the ZAP name (Z00005).

   Do not delete any ZAP names from the file unless you no longer want to apply them. The VMFZAP EXEC erases all text files that have been previously zapped for a product and then re-applies all ZAPs found in the ZAP list. Thus, you need to make sure that the ZAP list that you create or add to contains all the ZAPs that you want to apply.

See page N-15 for an example of a ZAP List.

5. Invoke the VMFZAP EXEC using the *prodid* parameter.

   VMFZAP erases all previously applied TEXT files from the ZAP disk. Then all the ZAPs listed in the *prodid* ZAPLIST (including Z00005) are applied to the product.

6. Finally, invoke the product-supplied EXEC that builds the executable version of the product.

Apply Corrective Service to Object Code

Corrective service is the application of a Program Temporary Fix (PTF) or an IBM Change Team supplied fix to correct a problem.

Corrective fixes can be:

- **Fixes on a PTF tape** provided by Program Support Services (PSS) or by the Change Teams. These fixes are the result of closed valid APARs.
- **ZAPs** provided by the Change Teams over the phone.
- **Relief fixes** provided by the Change Teams on a tape prior to APAR closure to fix severe problems.

Suppose you order PTF UV00007 for your product and that UV00007 affects modules A and B. Here's what you need to do:

1. Backup your existing system and verify that the backup copy you have is good.

2. Use VMFPLC2 to load the first physical file on the corrective service tape. (In the example, corrective service tape is shipped by PSS.) The first tape file contains information for you in the cover letter and in the Program Identification Number (PIN) pages.

3. Use VMFPLC2 again to load the second physical file on the tape. This tape file contains the following for UV00007:

   - A Service Control File
   - SCFs for any prerequisites
   - The actual fixes
   - The prerequisites.

   PTF UV00007 contains fixes for modules A and B, and these fixes are on the tape as CMS files A TXT00007 and B TXT00007.

   The fixes and the SCFs must be loaded to the delta disk. The other files on the tape can be loaded to any other disk.

4. Create an Apply List named FIX007 APPLIST. (You can specify any filename, but the filetype **must** be APPLIST.) You should save these lists so that you can use them later when you apply a PUT.

   It is in this Apply List that you specify the fixes that you want to merge into the product. List one PTF per line in the file.

   Note: You must list prerequisite PTFs before the PTFs that need them.

   See page N-12 for an example of an Apply List.

5. Invoke the VMFMERGE EXEC with the parameters PTFLIST FIX007. VMFMERGE will merge UV00007 into the product.

6. Now invoke the VMFZAP EXEC for the product. (The latest ZAP List for this product must still be available.) Issuing VMFZAP with the latest ZAP List cleans up superseded ZAPs and prevents regression of the corrective service just applied by old ZAPs.

   Note: If ZAP Z00005 was superseded by PTF UV00007, the ZAP will not be re-applied. If ZAP Z00005 was not superseded by PTF UV00007, the ZAP will be re-applied.

7. Finally, invoke the product-supplied service exec (or VMSERV) to build the executable version of the product.

Apply Preventive Service to Object Code

Preventive service is the application of Program Temporary Fixes (PTFs) available on a Program Update Tape (PUT) in order to avoid known problems.

VMSERV is an exec procedure included on the PUT to help you when you apply service. Suppose you receive a PUT for your product. Here's what you need to do:

1. Use VMFPLC2 to load the first physical file on the preventive service tape. This file contains the PUT DOCUMENT. You should read this document because it contains information about the PUT.

2. Invoke VMSERV to apply the PUT. You are given the choice of loading the merge file, the delta file, or both. (It is fastest to load just the merge file. However, if you know there is a bad PTF in the merge file, then you need to load the delta file. Whether you should load the merge file or the delta file depends upon the severity of the bad PTF.) In this case, load the merge file. This gives you a new version of the Merge Log that includes all the PTFs that are now part of your product. See page N-7 for an example of a Merge Log.

3. Invoke VMFMERGE using the Apply Lists that were saved when corrective fixes were merged.

   If you apply any corrective service to your product, you should keep the Apply Lists that you used. During preventive service, when you invoke VMFMERGE, you use these same Apply Lists. This prevents you from losing any corrective fixes that were already merged but are not included on the PUT.

   For example, assume that FIX007 APPLIST is the only Apply List used. You invoke VMFMERGE with the parameter PTFLIST FIX007. If UV00007 is included in the PUT, then a message is displayed explaining that UV00007 has already been merged. If UV00007 is *not* included in the PUT, then a message is displayed telling you that UV00007 will be merged at this time.

4. Now invoke VMFZAP EXEC for the product. (You must still have the latest ZAP List for this product available.) Issuing VMFZAP with the latest ZAP List cleans up superseded ZAPs and prevents regression of the corrective service just applied by old ZAPs.

   Note: If ZAP Z00005 was superseded by PTF UV00007, the ZAP will not be re-applied. If ZAP Z00005 was not superseded by PTF UV00007, the ZAP will be re-applied.

5. Finally, invoke the product-supplied service exec (or VMSERV) to build the executable version of the product.

Merge Service

Merge a Single PTF (No Dependents or Supersedes)

To merge a PTF (i.e. UV00002) that does not have any corequisites and is not a prerequisite of any other PTF, here's the process:

Note: You must access the minidisk containing the **prodid** *VMFPARM file before you proceed.*

- You must have a Merge Log and Service Control File(s). You may also have a user Exclude List and a Reqby Log. Let's assume you have the Merge Log, Exclude List, and Service Control Files for product 5664167, as shown in the figures on the following pages.

```
:entry.UV00001  Merged   12/13/83 13:09:42 ELEM1 TEXT ELEM2 TEXT
```

Figure N-11. Merge Single PTF - Sample Merge Log for 5664167

```
*EXCLUDE list for product 5664167
```

Figure N-12. Merge Single PTF - Sample Exclude List for 5664167

```
:ptf.UV00001
:prodid.5664167
:changes.
 :element.ELEM1 TEXT
  :replace.TXTP0001
 :element.ELEM2 TEXT
  :replace.TXTP0002
:echanges.
:apartext.ELEM1 service fix
:apartext.ELEM2 service fix
```

Figure N-13. Merge Single PTF - Sample SCF for UV00001

```
:ptf.UV00002
:prodid.5664167
:changes.
 :element.ELEM3 TEXT
  :replace.TXTP0003
:echanges.
:apartext.ELEM3 service fix
```

Figure N-14. Merge Single PTF - Sample SCF for UV00002

- To merge UV00002, enter the following command:

  vmfmerge 5664167 ptf UV00002

- VMFMERGE:

  1. Reads the Merge Log and finds that UV00002 is not already Merged or Superseded.

  2. Reads the Exclude List and finds that UV00002 is not excluded either.

  3. Reads the Service Control Files and finds that ELEM3 TEXT is the element affected by UV00002.

     ELEM3 TEXT's replacement file (TXTP0003) is copied from the delta disk to the merge disk. TXTP0003 replaces the element and adds service history to ELEM3 TEXT.

  4. Updates the Merge Log to show that UV00002 has been merged.

The resulting Merge Log looks as follows. (The Service Control Files are not changed.)

```
:entry.UV00001  Merged   12/13/83 13:09:42 ELEM1 TEXT ELEM2 TEXT
:entry.UV00002  Merged   06/23/84 04:18:20 ELEM3 TEXT
```

Figure N-15. Merge Single PTF - Changed Merge Log for 5664167

Merge Multiple PTFs (with Dependents and Supersedes)

To merge more than one PTF (i.e. UV00004 and UV00005), each having a combination of prerequisites, corequisites, and supersedes, here's the process:

- You must have a Merge Log and Service Control Files. Let's assume you have the Merge Log and Service Control Files shown in the following figures.

```
:entry.UV00002  Merged   05/03/84 13:09:42 ELEM3 TEXT
```

Figure N-16. Merge Multiple PTFs - Sample Merge Log for 5664167

```
:ptf.UV00001
:prodid.5664167
:coreq.UV00003
:changes.
 :element.ELEM1 TEXT
  :replace.TXTP0001
 :element.ELEM2 TEXT
  :replace.TXTP0002
:echanges.
:apartext.ELEM1 service fix
:apartext.ELEM2 service fix
```

Figure N-17. Merge Multiple PTFs - Sample SCF for UV00001

```
:ptf.UV00002
:prodid.5664167
:changes.
 :element.ELEM3 TEXT
  :replace.TXTP0003
:echanges.
:apartext.ELEM3 service fix
```

Figure N-18. Merge Multiple PTFs - Sample SCF for UV00002

```
:ptf.UV00003
:prodid.5664167
:coreq.UV00001
:changes.
 :element.ELEM4 TEXT
  :replace.TXTP0004
 :element.ELEM5 TEXT
  :replace.TXTP0005
:echanges.
:apartext.ELEM4 service fix
:apartext.ELEM5 service fix
```

Figure N-19. Merge Multiple PTFs - Sample SCF for UV00003

```
:ptf.UV00004
:prodid.5664167
:prereq.UV00001
:changes.
 :element.ELEM6 TEXT
  :replace.TXTP0006
:echanges.
:apartext.ELEM6 service fix
```

Figure N-20. Merge Multiple PTFs - Sample SCF for UV00004

```
:ptf.UV00005
:prodid.5664167
:sup.UV00002
:changes.
 :element.ELEM3 TEXT
  :replace.TXTP003A
:echanges.
:apartext.ELEM3 service fix
```

Figure N-21. Merge Multiple PTFs - Sample SCF for UV00005

- To merge UV00004 and UV00005 using one command, you need to create an Apply List (see page N-12 for more details). Your Apply List may look like Figure N-22.

```
* This is my own apply list
UV00004  put on  UV00004
UV00005  put on  UV00005
```

Figure N-22. MYLIST APPLIST

- After you create the Apply List, enter the following command.:

  vmfmerge 5664167 ptflist mylist

- VMFMERGE:

  1. Looks for an Apply List with a filename of "mylist."

  2. Reads the Merge Log and finds that UV00004 and UV00005 are not merged or superseded. VMFMERGE also reads the Exclude List and finds that UV00004 and UV00005 are not excluded either.

3. Reads the Service Control Files and finds that:

   - UV00001 is a prerequisite of UV00004 and must be merged.
   - UV00003 is a corequisite of UV00001 and it too must be merged.
   - UV00002 is superseded by UV00005.

4. Determines that ELEM1 TEXT, ELEM2 TEXT, ELEM3 TEXT, ELEM4 TEXT, ELEM5 TEXT, and ELEM6 TEXT are the elements these PTFs affect.

   VMFMERGE copies these elements' replacement files from the delta disk and replaces the appropriate files on the merge disk. In addition, service history for these elements is added to the files.

5. Updates the Merge Log to show that UV00003, UV00001, UV0004, and UV00005 are merged and that UV00002 is superseded.

The resulting Merge Log is shown in Figure N-23 and the Reqby Log is shown in Figure N-24. (The Service Control Files are not changed.)

```
:entry.UV00002  Merged      05/03/84 09:12:42 ELEM3 TEXT
:entry.UV00003  Merged      09/25/83 12:43:17 ELEM4 TEXT ELEM5 TEXT
:entry.UV00001  Merged      09/25/83 12:43:17 ELEM1 TEXT ELEM2 TEXT
:entry.UV00004  Merged      09/25/84 12:43:17 ELEM6 TEXT
:entry.UV00005  Merged      09/25/84 12:43:17 ELEM3 TEXT
:entry.UV00002  Superseded  09/25/84 12:43:17 By UV00005
```

Figure N-23. Merge Multiple PTFs - Sample Changed Merge Log for 5664167

```
* Reqby Log for 5664167
:entry.UV00001 UV00003 UV00004
:entry.UV00003 UV00001
```

Figure N-24. Merge Multiple PTFs - Changed Reqby Log for 5664167

Remove Service

Remove a Single PTF (No Dependents or Supersedes)

To remove a PTF (i.e. UV00008) that does not have any corequisites and is not a prerequisite of any other PTF, here's the process.

- You must have a Merge Log and Service Control Files. Let's assume you have the Merge Log and Service Control Files shown in the following figures.

```
:entry.UV00001  Merged      12/13/83 13:09:42 ELEM1 TEXT
:entry.UV00004  Merged      12/15/83 06:28:17 ELEM4 TEXT ELEM5 TEXT
:entry.UV00005  Merged      12/15/83 06:28:17 ELEM5 TEXT ELEM1 TEXT
:entry.UV00006  Merged      01/12/84 17:56:39 ELEM6 TEXT ELEM1 TEXT
:entry.UV00001  Superseded  01/12/84 17:56:39 By UV00006
:entry.Z00001   Superseded  01/12/84 17:56:39 By UV00006
:entry.UV00007  Merged      01/12/84 17:56:39 ELEM6 TEXT
:entry.UV00008  Merged      01/12/84 17:56:39 ELEM7 TEXT
:entry.UV00009  Merged      01/12/84 17:56:39 ELEM8 TEXT
```

Figure N-25. Remove Single PTF - Merge Log for 5664167

```
:ptf.UV00001
:prodid.5664167
:changes.
 :element.ELEM1 TEXT
  :replace.TXTP0001
:echanges.
```

Figure N-26. Remove Single PTF - SCF for UV00001

```
:ptf.UV00004
:prodid.5664167
:prereq.UV00001
:changes.
 :element.ELEM4 TEXT
  :replace.TXTP0004
 :element.ELEM5 TEXT
  :replace.TXTP0004
:echanges.
:apartext.PP00004 - Split DMKABN for new base register
:apartext.PP00009 - Update CP loadlist to add DMKNEW
```

Figure N-27. Remove Single PTF - SCF for UV00004

```
:ptf.UV00005
:prodid.5664167
:prereq.UV00004
:coreq.UV00006
:changes.
 :element.ELEM5 TEXT
  :replace.TXTP0005
:echanges.
```

Figure N-28. Remove Single PTF - SCF for UV00005

```
:ptf.UV00006
:prodid.5664167
:coreq.UV00005
:sup.UV00001 Z00001
:changes.
 :element.ELEM1 TEXT
  :replace.TXTP0006
 :element.ELEM6 TEXT
  :replace.TXTP0006
:echanges.
```

Figure N-29. Remove Single PTF - SCF for UV00006

```
:ptf.UV00007
:prodid.5664167
:prereq.UV00006
:changes.
 :element.ELEM6 TEXT
  :replace.TXTP0007
:echanges.
```

Figure N-30. Remove Single PTF - SCF for UV00007

```
:ptf.UV00008
:prodid.5664167
:prereq.UV00007
:changes.
 :element.ELEM7 TEXT
  :replace.TXTP0008
:echanges.
```

Figure N-31. Remove Single PTF - SCF for UV00008

```
:ptf.UV00009
:prodid.5664167
:prereq.UV00004
:changes.
 :element.ELEM8 TEXT
  :replace.TXTP0009
:echanges.
```

Figure N-32. Remove Single PTF - SCF for UV00009

- To remove UV00008, enter the following command:

  vmfremov 5664167 ptf UV00008

- VMFREMOV:

  1. Uses the Merge Log and the Service Control Files to build the Reqby Log shown in the following figure. Notice that there is an entry for each PTF which is a prerequisite or corequisite of another PTF which has a entry of *merged* in the Merge Log.

```
* Reqby Log for 5664167
:entry.UV00005 UV00006
:entry.UV00006 UV00005 UV00007
:entry.UV00001 UV00004
:entry.UV00004 UV00005 UV00009
:entry.UV00007 UV00008
```

Figure N-33. Remove Single PTF - Sample Reqby Log for 5664167

  2. Reads the Reqby Log and finds that UV00008 does not have any dependent PTFs that must also be removed.

  3. Reads the Merge Log and finds that ELEM7 TEXT is the only element affected by UV00008.

  4. Erases ELEM7 TEXT from the merge disk. Since, there are no other previously merged PTFs affected, the text deck for ELEM7 is taken from the base disk with the next build.

  5. Updates the Merge Log to show that UV00008 is removed and is no longer merged.

  6. Removes UV00007 from the Reqby Log since there are no longer any dependents (i.e.UV00008) for that PTF.

The resulting Merge Log is shown in Figure N-34 and the resulting Reqby Log is shown in Figure N-38 on page N-31. (The Service Control Files are not changed.)

```
:entry.UV00001  Merged      12/26/83 13:09:42 ELEM1 TEXT
:entry.UV00004  Merged      12/28/83 06:28:17 ELEM4 TEXT ELEM5 TEXT
:entry.UV00005  Merged      12/28/83 06:28:17 ELEM5 TEXT
:entry.UV00006  Merged      01/22/84 17:56:39 ELEM6 TEXT ELEM1 TEXT
:entry.UV00001  Superseded  01/22/84 17:56:39 By UV00005
:entry.Z00001   Superseded  01/22/84 17:56:39 By UV00005
:entry.UV00007  Merged      01/22/84 17:56:39 ELEM6 TEXT
*entry.UV00008  Merged      01/22/84 17:56:39 ELEM7 TEXT
:entry.UV00009  Merged      01/22/84 17:56:39 ELEM8 TEXT
*entry.UV00008  Removed     02/02/84 17:56:39
```

Figure N-34. Remove Single PTF - Changed Merge Log for 5664167

The changed entries in the Merge Log are now comments; therefore, they are preceded by an asterisk (*).

```
* Reqby Log for 5664167
:entry.UV00005 UV00006
:entry.UV00006 UV00005 UV00007
:entry.UV00001 UV00004
:entry.UV00004 UV00005 UV00009
```

Figure N-35. Remove Single PTF - Changed Reqby Log for 5664167

The Reqby Log no longer contains the line **:entry.UV00007 UV00008**.

Remove Multiple PTFs (with Dependents and Supersedes)

To remove multiple PTFs (i.e. UV00005 and UV00009), each having a combination of prerequisites, corequisites, and supersedes, here's the process:

- You must have a Merge Log and Service Control Files. Let's assume you have the same Merge Log shown in Figure N-34 on page N-29, the Reqby Log shown in Figure N-35 on page N-29 and the Service Control Files shown in Figure N-26 on page N-26 through Figure N-32 on page N-28.

- To remove UV00005 and UV00009 using one command, you need to create a Remove List (see page N-13 for more details). Your Remove List may look like like Figure N-36.

```
* This is my own remove list
UV00005  take off UV00005
UV00009  take off UV00009
```

Figure N-36. MYLIST REMLIST

- After you create the Remove List, enter the following command:

  vmfremov 5664167 ptflist mylist

- VMFREMOV:

  1. Looks for the file MYLIST REMLIST on one of the delta disks, because you entered the keyword **ptflist** followed by a listname (**mylist**).

  2. Reads the Reqby Log and finds that to remove UV00005, UV00006 and UV00007 must also be removed.

     Since UV00001 and Z00001 are superseded by UV00005 (which now needs to be removed), UV00001's status is changed back to merged and Z00001 is no longer superseded.

     UV00009 has no dependents, so it can be removed without removing and other PTFs.

  3. Erases from the merge disk, or replaces with other levels from the delta disk, the following elements: ELEM5 TEXT, ELEM1 TEXT, ELEM6 TEXT,and ELEM8 TEXT.

  4. Updates the Merge Log to show that UV00005, UV00006, UV00007, and UV00009 are removed and are no longer merged. In addition, UV00001 is no longer superseded and now has a status of merged. Z00001 is no longer superseded.

  5. Updates the Reqby Log by removing any entries for the removed PTFs and eliminates any dependents for those PTFs.

The resulting Merge Log is shown in Figure N-37 on page N-31 and the resulting Reqby Log is shown in Figure N-38 on page N-31. (The Service Control Files are not changed.)

```
:entry.UV00001  Merged      01/13/84 13:09:42 ELEM1 TEXT
:entry.UV00004  Merged      01/13/84 06:28:17 ELEM4 TEXT ELEM5 TEXT
*entry.UV00005  Merged      01/13/84 06:28:17 ELEM5 TEXT ELEM1 TEXT
*entry.UV00006  Merged      02/12/84 17:56:39 ELEM6 TEXT ELEM1 TEXT
*entry.UV00001  Superseded  02/12/84 17:56:39 By UV00005
*entry.Z00001   Superseded  02/12/84 17:56:39 By UV00005
*entry.UV00007  Merged      02/12/84 17:56:39 ELEM6 TEXT
*entry.UV00008  Merged      02/12/84 17:56:39 ELEM7 TEXT
*entry.UV00009  Merged      02/12/84 17:56:39 ELEM8 TEXT
*entry.UV00008  Removed     02/22/84 05:56:39
*entry.UV00007  Removed     02/22/84 05:56:39
*entry.UV00006  Removed     02/22/84 05:56:39
*entry.UV00005  Removed     01/23/84 06:28:17
*entry.UV00009  Removed     02/22/84 05:56:39
```

Figure N-37. Remove Multiple PTFs - Changed Merge Log for 5664167

The changed entries in the Merge Log are now comments; therefore, they are preceded by an asterisk (*).

```
* Reqby Log for 5664167
:entry.UV00001 UV00004
```

Figure N-38. Remove Multiple PTFs - Changed Reqby Log for 5664167

Prevent Regression

There are precautions you can take to ensure that the level of service applied to your product does not regress. You should:

- Save Apply Lists

  Whenever you merge corrective fixes to your product, you **must** save the Apply Lists you use. You use these lists after you apply preventive service.

- Re-apply Corrective Fixes

  If you have merged any corrective fixes into your product, you should invoke VMFMERGE after applying the PUT. You should do this for each Apply List you used when applying corrective service. You should apply corrective fixes in chronological order, but you do not have to.

  Note: You can create one large Apply List that contains all the other Apply Lists and invoke VMFMERGE using just the large Apply List.

- Re-apply ZAPs

  You can invoke VMFZAP over and over without harming your product. Each time you do, it erases all ZAP TEXT files that were previously ZAPped. Then all the ZAPs (except those that have been superseded) listed in *prodid* ZAPLIST are re-applied.

  You should invoke VMFZAP just before you invoke the product supplied exec that builds the executable version of the product.

Remove a Fix-in-Error

Let's assume that:

PUT 1	contains PTF 10	which affects	Module A
PUT 2	contains PTF 11	which affects	Module A
	contains PTF 12	which affects	Module B
	contains PTF 13	which affects	Module C
PUT 3	contains PTF 14	which affects	Module B and C
PUT 4	contains PTF 15	which affects	Module A

Now, after you apply PUT 4, suppose you find an error in your product after it has been built. When you report the error to your Support Center, they tell you the filenames of the PTF(s) in error that must be 'backed off'. For example, if PTF 12 is bad and you want to remove it, here's what you need to do:

1. Invoke VMFREMOV specifying the prodid and PTF12. VMFREMOV removes PTF12 and PTF14.

2. Invoke VMFZAP specifying the prodid to apply any previously applied ZAPs. If the ZAP List has not been changed, then VMFZAP reapplies the ZAPs which were superseded by the changed just removed.

3. Finally, invoke the product-supplied exec that builds the executable version of the product.

Summary of Changes

To obtain editions of this publication pertaining to earlier releases of VM/SP, you must order using the pseudo-number assigned to the respective edition. For:

Release 4, order ST00-1592

Release 3, order SQ24-5237

Release 2, order ST00-0916

Release 1, order SQ19-6201.

Summary of Changes
for SN24-5761-0
for VM/SP Release 5.0

Added:

- Documentation of the PASSMOD EXEC that changes logon passwords for optional feature program product userids in the CP directory file, added to Chapter 8.
- Table containing shared segment information for optional feature program products, added to Chapter 9.
- PASSMOD panel added to Chapter 10.
- Message descriptions added to Chapter 10.

Changed:

- Appendix L updated and reorganized as Chapter 8.
- Appendix M updated and reorganized as Chapter 9.
- INSTFPP and DIRECGEN panels moved from Appendix M to Chapter 10.

Deleted:

- Section on Automatic Logon of Disconnected Virtual Machines in Appendix M.

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.
- Chapter 1, " 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 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 Discontiguous 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.

- "Using the PASSMOD EXEC to Change 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 SN24-5751-0
for VM/SP Release 4.0

Added:

- Documentation of the DIRECGEN EXEC that sets up the CP directory file for optional feature program products available in the System Offering, added to Appendix L.
- Program product installation overview, added to Appendix L.
- INSTFPP and DIRECGEN panels added to Appendix L.
- Table containing product directory information for the optional feature program products, added to Appendix M.

Changed:

- Chapter 6 updated and reorganized as Appendix L.
- Appendix I updated and reorganized as Appendix M.

Deleted:

- Table of DASD requirements in Appendix I.
- Section on the PPPREP EXEC in Chapter 6.

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 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

Changed: 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.

directory. In VM/SP, a CP disk file that defines each virtual machine's normal configuration.

display device. An I/O device that gives a visual representation of data.

display terminal. A terminal with a component capable of displaying information on a viewing surface such as a cathode ray tube (CRT) or gas panel.

dump. To write the contents of part or all of main storage, or part or all of a minidisk, to an external medium such as tape.

E

EC level. Engineering change level.

ECMODE. A mode in which all the features of a System/370 computing system, including dynamic address translation, are operational.

element. In reference to installation and service of a product, a file provided on a product tape or a program update tape (PUT) as input to the build process (see *build*). An element is the smallest serviceable unit of a component. There may be several files associated with a given element, and each file has the same filename.

emulation. The use of programming techniques and special machine features to permit a computing system to execute programs written for another system.

Emulation Program (EP). A control program that allows an IBM 3704 or 3705 Communications Controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control.

entry point. The address or label of the first instruction executed upon entering a computer program, a routine, or a subroutine.

EP. See *Emulation Program (EP).*

EREP file. A collection of error records produced by the malfunction of hardware components on a computer and stored for processing by the Environmental Recording, Editing, and Printing (EREP) program.

Exclude List. A file listing Program Temporary Fixes (PTFs) to be omitted from a product.

F

FB-512 device. A fixed-block architecture device that stores data in 512-byte blocks. See also *fixed-block architecture (FBA) device.*

FBA. See *fixed-block architecture (FBA) device.*

FCB. See *forms control buffer (FCB).*

filename. A 1-to-8-character alphameric filed, comprised of A-Z, 0-9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

fixed-block architecture (FBA) device. A disk storage device that stores data in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the particular file.

forms control buffer (FCB). In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output.

free storage. Blocks of memory that are available for temporary use by programs or by the system.

full screen editor. An editor used at a display terminal where an entire screen of data is displayed at once and where the user can access the data through commands or by using a cursor.

H

history files. One or more CMS files that describe the changes (with a date and time stamp) made to the VM/SP system and its installed software products.

I

IML. See *initial microprogram load (IML).*

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 or other computer program to commence operation.

Installation Verification Procedure (IVP). A program that verifies the correct installation of VM/SP by testing various CP and CMS functions.

I/O. input/output

IPL. See *initial program load (IPL).*

IVP. See *Installation Verification Procedure (IVP).*

L

load map. A file containing the storage addresses of the control sections and entry points of a program loaded into storage.

loader. A routine, commonly a computer program, that reads data into main storage.

local update. A user's own modifications or enhancements to VM/SP modules or source code.

K

K. See *kilobyte (K).*

kilobyte (K). 1024 bytes

M

M. See *megabyte (M).*

MACLIB. See *macro library (MACLIB).*

macro. See *macro definition* and *macro instruction.*

macro definition. A set of statements that defines the name of, format of, and conditions for generating a sequence of assembler language statements from a single source statement.

macro instruction. In assembler language programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macro instruction in the program.

macro library (MACLIB). A collection of macro definitions and/or copy files.

MDISK. Another name for minidisk. It is also the directory statement used to define a minidisk.

megabyte (M). 1,048,576 bytes.

merge. In reference to service, including a change (a PTF) into the code.

merge disk. In reference to service of a product, the disk containing changes that have been applied to a product. These changes have been copied from the delta disk and renamed to match the corresponding file on the base disk.

Merge Log. A file (log) of changes which have been merged and/or superseded.

module. A unit of a software product that is discretely and separately identifiable with respect to modifying, compiling, merging with other units, and/or loading and executing; for example, the input to, or output from, a compiler, assembler, linkage editor, or executive routine.

MP. See *multiprocessor (MP).*

multiprocessor (MP). A computer configuration that employs two or more processors under integrated control.

N

node. In a network, a point where one or more functional units interconnect transmission lines.

nucleus. That part of a control program that is resident in main storage.

O

object code. Compiler or assembler output that is itself executable machine code or is suitable for further processing to produce executable machine code.

object maintained. Service which is done through replacement of an element.

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.

overlay. The technique of repeatedly using the same areas of internal storage during different stages of a program.

P

pack. A set of flat circular recording surfaces used in a disk storage device. A disk pack.

page. A fixed-length block that has a virtual address and that can be transferred between real storage and auxiliary storage.

paging. The process of transferring pages between real storage and external page storage.

parameter. A variable that is given a constant value for a specific application and that may denote the application.

prerequisite change. A change that must be applied to the system before another change can be applied. For example, Change2 lists Change1 as a prerequisite. This indicates that you must apply Change1 before you apply Change2.

preventive service. The process of loading the contents of a program update tape (PUT) to minidisks, and applying all changes. The last step of preventive service is to perform the build process (see build).

printer universal character set. A printer feature that permits the use of a variety of character arrays.

process. A systematic sequence of operations to produce a specified result. A process is usually logical, not physical.

product. Any separately installable software program, whether supplied by IBM or otherwise, that is distinct from others and is recognizable by a unique identification code. Common examples of software products include:

5664-167 - Virtual Machine/System Product
5748-FO3 - VS/FORTRAN Program Product
5748-RC1 - VM/Pass-through Product

The product identification code is unique to a given product, but does not identify the release level of that product.

Product Parameter File. A file containing records that identify the various product minidisk addresses used for installation and service.

program status word (PSW). An area in storage used to indicate the order in which instructions are executed, and to hold and indicate the status of the computer system.

Program Temporary Fix (PTF). Code changes needed to correct a problem reported in an APAR. The corrected code is included in later releases. It includes element replacements (for object code) or element updates (for source code) for elements changed by the fix. It also defines limitations on when the PTF can be included. Each PTF is unique to a given release of a product. If the same problem occurs in multiple releases of a product, a separate PTF is defined for each release. A PTF defines only one replacement or update for each element, regardless of how many APARs are fixed.

prompt. A displayed message that assists a terminal operator by describing required input or by giving operational information.

PSW. See *program status word (PSW).*

PTF. See *program temporary fix (PTF).*

PTF number. A number assigned by service organizations that uniquely identifies a PTF; for example, IBM uses UVnnnnn for a VM-unique product, and UPnnnnn for a cross-system product. PTFs to different products or to different releases of a product have different numbers.

program update tape (PUT). A tape containing customized service. Each service tape contains cumulative service for the customer's products back to the earliest release level of the product still supported. The tape is distributed to authorized customers of the products at scheduled intervals.

PUT. See *program update tape (PUT).*

R

regression. The act of causing elements to go back to earlier levels. This may occur when applying changes from a program update tape (PUT) to elements updated by corrective service or user modifications.

Remove List. A file listing the Program Temporary Fixes (PTFs) you want to remove from a product.

Reqby Log. A file (log) of all dependent Program Temporary Fixes (PTFs) of each PTF which has a merged or superseded entry in the merge log.

S

SCF. See *service control file (SCF).*

service control file (SCF). A file that describes a PTF, and that contains all the information needed to install the PTF.

Service Log. A file containing information about service that has been applied to a product. This file is the Merge Log with the ZAP Log appended on the end.

service tape. A tape that contains service for those IBM-supplied software products for which the customer is licensed. Distributed at scheduled intervals, and cumulative back to the earliest release level still available from IBM Software Distribution (ISD), the tape contains a collection of PTFs for selected program products.

source code. A program that has not yet been compiled or assembled to produce the corresponding object code.

source maintained. Service through the updating of an element.

source update. A change to the original assembler code provided with a product. VM source code is contained in files with a filetype of ASSEMBLE. To update an assemble file you create update files containing control statements that describe the changes that are to be made.

stand-alone program. A program that is independent of any other program or system.

superseded. In reference to service, a change which is no longer needed (some later fix has replaced it) is called superseded.

T

TSAF. See *Transparent Services Access Facility (TSAF).*

Transparent Service Access Facility (TSAF). An optional component of VM/SP. TSAF lets users connect to and communicate with local or remote virtual machines within a group of systems. With TSAF, you can connect to a program by specifying a name that the program has made known, instead of specifying a userid or nodeid.

U

uniprocessor (UP). A computer configuration that consists of a single processor or that uses only one processor of an attached processor (AP) or multiprocessor (MP) system.

universal character set. See *printer universal character set.*

UP. See *uniprocessor (UP).*

update file. The actual changed code to be used by the assembler or compiler to produce new object code. This file exists only in source-maintained products.

update number. A number assigned by service organizations and uniquely associated with a PTF. The PTF number and update number may be the same. If they are different, there is a one-to-one correspondence between the PTF number and the update number. With VM source-maintained products, only one APAR is fixed in each PTF, and the PTF number is not the same as the update number. The update number format is *raaaaacc*, where *r* identifies the product and release, *aaaaa* is the APAR number, and *cc* defines the component.

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.

userid. A 1-to-8-character alphameric symbol that identifies a terminal user.

W

warm start. The result of an IPL that does not erase previous system data. 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.

Z

ZAP. ZAP is a CMS command that changes or dumps MODULE, LOADLIB, or TXTLIB files. It may be used to change either fixed or variable length MODULE files. It is for use by system support personnel only.

ZAP List. A file listing ZAPs to be applied to a product. If no ZAPs are applied then you do not need to create a ZAP list.

ZAP Log. A file (log) of applied ZAPs. Each entry has information on the date and time that the change was processed and a list of the elements (TEXT files) that the change affects.

Related Publications

This section lists IBM books that can help you plan and install your system. The *VM/SP Library Guide, Glossary, 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, and 4300 Processors Bibliography*, GC20-0001.

- **Planning**

  Virtual Machine/System Product Release 5:

  Release 5 Guide, SC24-5290
  Planning Guide and Reference, SC19-6201
  Transparent Services Access Facility Reference, SC24-5287

  Remote Spooling Communications Subsystem Networking Version 2:

  Planning and Installation, SH24-5057

  Other Non-VM/SP titles:

  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 - 5742-017) SCP 5742, 5744-AN1/BA2, 5747-AG1/AJ2, GC30-3008

- **Installation**

  Remote Spooling Communications Subsystem Networking Version 2:

  Planning and Installation, SH24-5057

  Other Non-VM/SP titles:

  IBM 3850 Mass Storage System (MSS) Installation 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 5:

  Terminal Reference, GC19-6206
  Introduction, SC19-6200
  Operator's Guide, SC19-6202
  System Messages and Codes, SC19-6204
  CMS Command Reference, SC19-6209
  CP Command Reference, SC19-6211
  CMS for System Programming, SC24-5286
  CP for System Programming, SC24-5285

  Virtual Machine:

  Diagnosis Guide, LY24-5241
  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

VM/VTAM

VM/VTAM Communication Network Application Messages, GC27-0510
VM/VTAM Communication Network Application Logic, LY38-3033

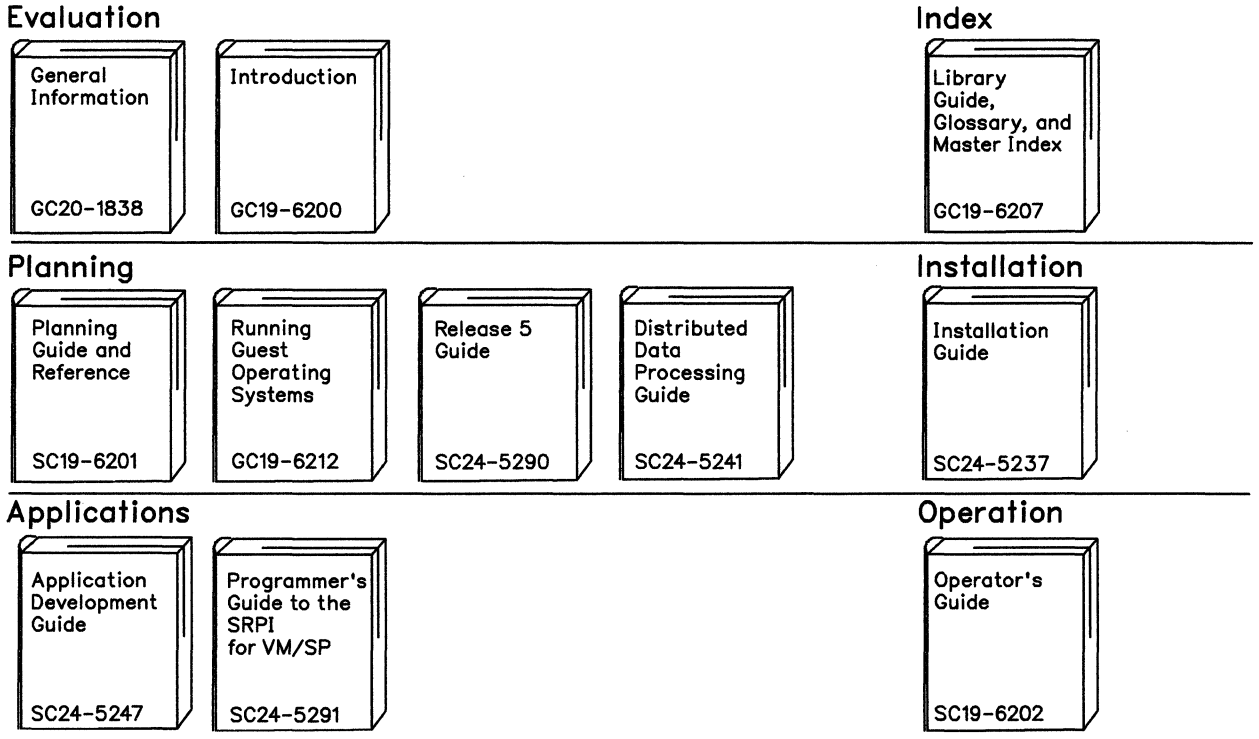
EREP

EREP User's Guide and Reference, GC28-1378

Miscellaneous

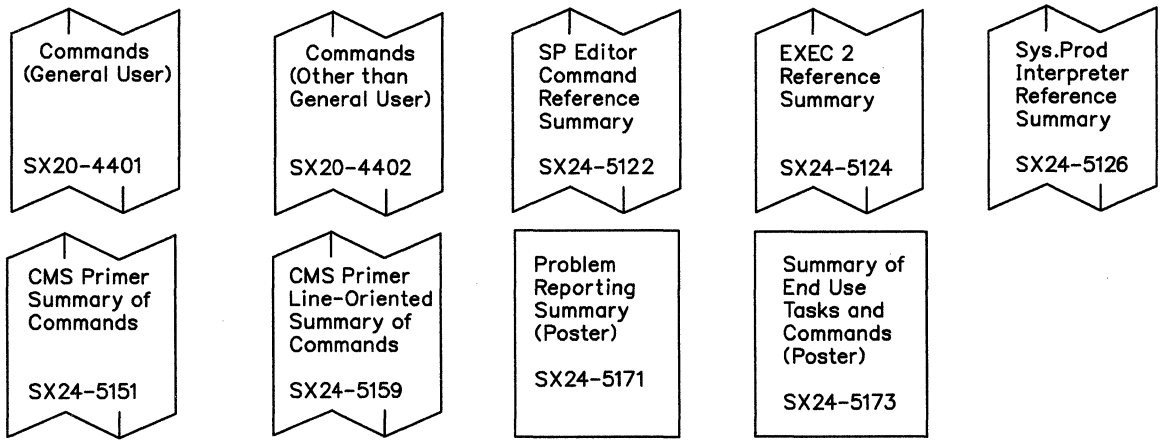
3270 Information Display System Library User's Guide, GA23-0058
IBM OS/VS Linkage Editor and Loader, GC26-3813
Input/Output Configuration Program User's Guide and Reference, GC28-1027
Device Support Facilities User's Guide and Reference, GC35-0033

The VM/SP Library (Part 1 of 3)



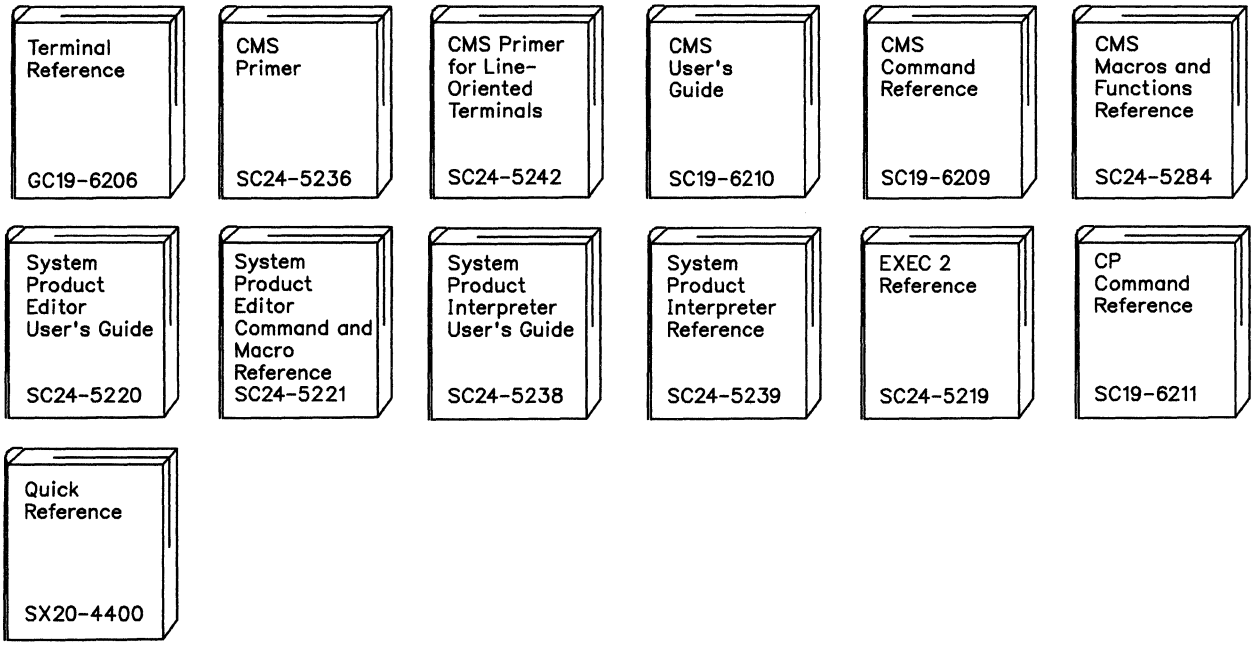
Reference Summaries

To order all of the Reference Summaries, use order number SBOF-3242

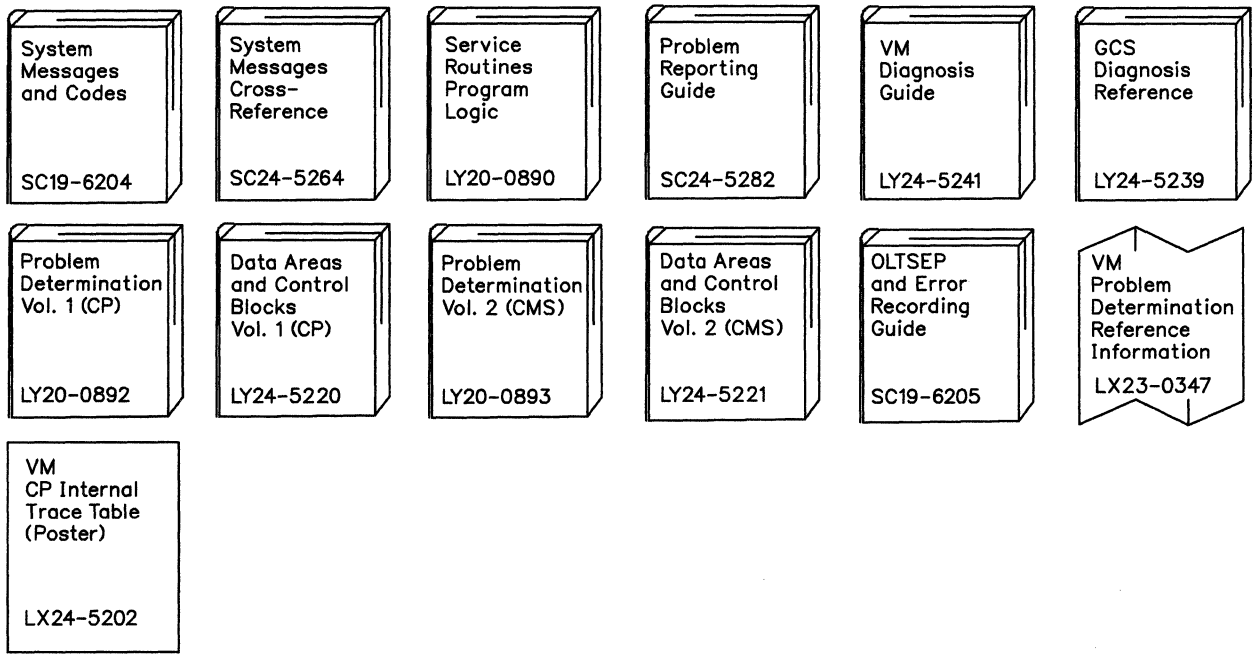


The VM/SP Library (Part 2 of 3)

End Use



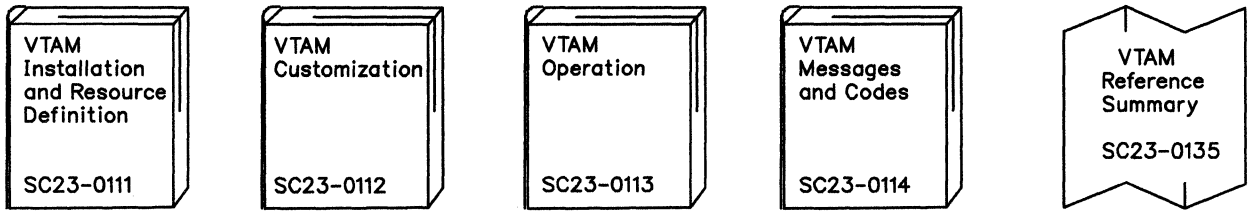
Diagnosis

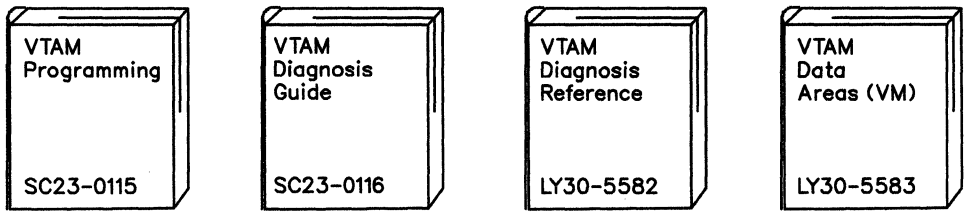


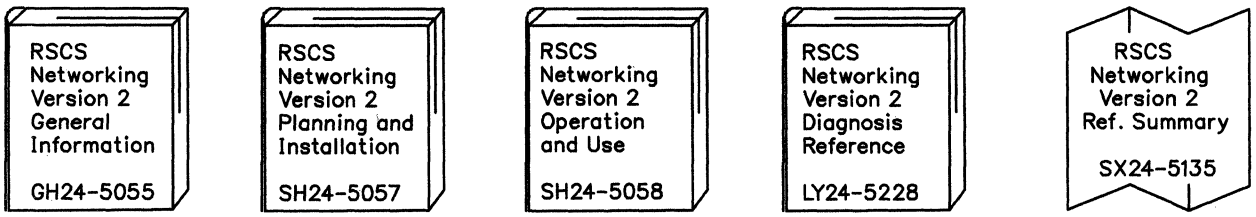
The VM/SP Library (Part 3 of 3)

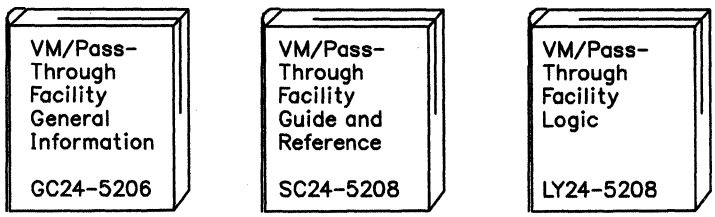
Administration

Auxiliary Communication Support









Special Characters

A

B

C

Index

Special Characters

A

B

C

D

E

F

G

H

K

L

O

P

Q

R

S

T

U

V

Y

Z

Numerics

International Business Machines Corporation
P.O. Box 6
Endicott, New York 13760

File No. S370/4300-34
Printed in U.S.A.

SC24-5237-3

SN24-5761-00

VM/SP Installation Guide
Order No. SC24-5237-3

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.

SC24-5237-3

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®

VM/SP Installation Guide
Order No. SC24-5237-3

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.

SC24-5237-3

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®