Virtual Machine/
Enterprise Systems Architecture

Service Guide

Version 2 Release 4.0

GC24-5838-01

Virtual Machine/
Enterprise Systems Architecture

Service Guide

Version 2 Release 4.0

GC24-5838-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xi.

| **Second Edition (July 1999)**

| This edition applies to Version 2, Release 4, Modification 0 of IBM® Virtual Machine/Enterprise Systems Architecture (VM/ESA®)
| (product number 5654-030) and to all subsequent releases and modifications until otherwise indicated in new editions.

| This edition replaces GC24-5838-00.

Contents

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10594-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks and Service Marks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

BookManager
DFSMS/VM
ECKD
ES/3090
ES/4381
IBM
IBMLink
ES/9000
ESA/390
Language Environment
Library Reader
NetRexx
OpenEdition
RACF
S/390
SP
SP2
System 370
VM/ESA
VTAM

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This book describes how to service Virtual Machine/Enterprise Systems Architecture (VM/ESA) Version 2 Release 4.0.

Who Should Read This Book

This book is intended for system programmers and anyone else responsible for maintaining a VM/ESA system.

What You Should Know before Reading This Book

This book assumes that you understand the concepts of VM/ESA service and are familiar with the syntax and usage of VM/ESA service tools. **Before using this book, you should read the** *VM/ESA: VMSES/E Introduction and Reference***, which presents these concepts and describes the VM/ESA service tools.**

What This Book Contains

This book contains the Product Service Upgrade (PSU) service procedure, which utilizes the VM/ESA Recommended Service Upgrade (RSU), and the corrective (COR) service procedure for all VM/ESA components. Local service and modifications as well as reworking local service is also discussed.

How To Use This Book

The procedures in this book are in a two-column format. The left column shows the representative sequence of user entries and some system responses, and the right column contains comments and instructions about the entries shown in the left column.

Where to Find More Information

See the bibliography at the back of this publication for more information concerning VM/ESA.

For more information on the service tools provided by VMSES/E (besides reviewing the *VM/ESA: VMSES/E Introduction and Reference*) use the online VM/ESA HELP facility. For example, to display a menu of VMSES/E commands, enter:

```
help vmses menu
```

To display information about a specific VMSES/E command (VMFREC in this example), enter:

```
help vmses vmfrec
```

For more information about using the HELP Facility, see the *VM/ESA: CMS User's Guide* . To display the main HELP Task Menu, enter:

```
help
```

For more information about the HELP command, see the *VM/ESA: CMS Command Reference* or enter:

```
help cms help
```

Roadmap for Installing, Servicing, or Migrating to VM/ESA

If you are installing or servicing VM/ESA Version 2 Release 4.0, or migrating to VM/ESA Version 2 Release 4.0 from a previous VM release, you should be familiar with the following documents. You should refer to them in the order they are listed.

Procedure	Document
Installation	1. ***VM/ESA Program Directory:*** Contains relevant program, APAR, and service level information and is shipped with the VM/ESA product. If necessary may contain the latest information regarding VM/ESA product installation. Helps you plan and obtain the information that is required during your service and installation tasks. 2. ***VM/ESA: Planning and Administration:*** Helps you estimate the DASD storage that is required for the CMS virtual machines, SFS and CRR. Details reference information on CP system definition files. 3. ***VM/ESA: Installation Guide:*** Steps you through how to install a VM/ESA Version 2 Release 4.0 operating system. 4. ***VM/ESA: VMSES/E Introduction and Reference:*** Introduces you to the VMSES/E installation tools that are used during a VM/ESA installation and provides information on planning and managing Software Inventory Tables. It also contains reference information on VMSES/E and a summary of what is new in this release. 5. ***VM/ESA: CMS File Pool Planning, Administration, and Operation:*** Provides guidance and reference information about planning for and managing CMS file pools. Provides specific information on the Shared File System (SFS) and the Coordinated Resource Recovery (CRR) facility. File pool administration and operation commands and start-up parameters are discussed. Also describes how to tailor the SFS defaults. 6. ***VM/ESA: System Messages and Codes:*** Provides messages, codes, and restrictions when using VM/ESA. Conditions that generate these messages and codes are explained, the resulting actions are described, and proper responses are suggested.
Service	1. ***VM/ESA: Service Guide:*** Provides service procedures used for VM/ESA components and is designed to be used with the *VM/ESA: VMSES/E Introduction and Reference*. It also shows you how to manage your Software Inventory Tables while servicing VM. Information on local service and removing service is also discussed. 2. ***VM/ESA: VMSES/E Introduction and Reference:*** Introduces you to the VMSES/E service tools that are used while servicing your VM/ESA system and provides information on planning and managing Software Inventory Tables. It also contains reference information on VMSES/E and a summary of what is new in this release. 3. ***VM/ESA: System Messages and Codes:*** Provides messages, codes, and restrictions when using VM/ESA. Conditions that generate these messages and codes are explained, the resulting actions are described, and proper responses are suggested.
Migration	1. ***VM/ESA: Conversion Guide and Notebook:*** Describes strategies for planning your conversion to VM/ESA Version 2 Release 4.0 from VM/ESA Version 1 Release 1.5 370 Feature, VM/ESA Version 1 Release 2.0, VM/ESA Version 1 Release 2.1, VM/ESA Version 1 Release 2.2, VM/ESA Version 2 Release 1.0, or VM/ESA Version 2 Release 2.0. It compares the functions in your current system to the functions in VM/ESA Version 2 Release 4.0 and describes differences in the areas of installation, service, administration, system operation, virtual machine operation, application programming, and diagnosis. **Note:** If you are converting to VM/ESA Version 2 Release 4.0 from VM/SP 5 or 6, VM/SP HPO 5 or 6, or VM/ESA Version 1 Release 1.0 (370 Feature), you may want to obtain a copy of the *VM/ESA: Conversion Guide and Notebook for VM/SP, VM/SP HPO, and VM/ESA 370 Feature*. If you are converting to VM/ESA Version 2 Release 4.0 from VM/XA SP2 or 2.1, VM/ESA Version 1 Release 1.0 (ESA Feature), or VM/ESA Version 1 Release 1.1, you may want to obtain a copy of the *VM/ESA: Conversion Guide and Notebook for VM/XA SP and VM/ESA*. Each of these books is available at the VM/ESA Version 2 Release 1.0 level. You may also want to obtain a copy of the *VM/ESA: CMS Application Migration Guide*, available at the VM/ESA Version 1 Release 2.2 level. These books are not supplied with VM/ESA Version 2 Release 4.0. 2. ***VM/ESA: REXX/EXEC Migration Tool for VM/ESA:*** Helps you plan your migration by estimating the amount of effort required to migrate certain types of files to VM/ESA Version 2 Release 4.0 from VM/SP 5, VM/SP HPO 5, VM/XA SP2 or 2.1, VM/ESA Version 1 Release 1.0 (370 Feature), VM/ESA Version 1 Release 1.0 (ESA Feature), VM/ESA Version 1 Release 1.1, VM/ESA Version 1 Release 1.5 370 Feature, VM/ESA Version 1 Release 2.0, VM/ESA Version 1 Release 2.1, VM/ESA Version 1 Release 2.2, VM/ESA Version 2 Release 1.0, or VM/ESA Version 2 Release 2.0. 3. Installation documentation described above. 4. Service documentation described above. 5. ***VM/ESA: System Messages and Codes:*** Provides messages, codes, and restrictions when using VM/ESA. Conditions that generate these messages and codes are explained, the resulting actions are described, and proper responses are suggested.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other VM/ESA documentation:

- Visit our home page at:

  `http://www.ibm.com/s390/vm`

  There you will find the feedback page where you can enter and submit your comments.

- Send your comments by electronic mail to one of the following addresses:

Format	Address
Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL
IBMLink™	GDLVME(PUBRCF)

  Be sure to include the name of the book, the form number (including the suffix), and the page, section title, or topic you are commenting on.

- Fill out the form at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

Summary of Changes

This section describes the technical changes made in this edition of the book and in previous editions. For your convenience, the changes made in this edition are identified in the text by a vertical bar (|) in the left margin. This edition may also include minor corrections and editorial changes that are not identified.

How to Obtain Previous Editions of This Book

Previous editions of this book and other books in the VM/ESA library can be ordered using the order numbers listed in the *VM/ESA: General Information* manual. That book lists the order numbers and suffixes for VM/ESA books, as well as certain related books, for currently supported VM/ESA releases. When ordering a previous edition of any book, it is important to specify the correct order number suffix.

Summary of Changes for VM/ESA Version 2 Release 4.0

New Function

- ***VMFREM:*** A new exec, VMFREM, lets you remove (back out) service levels or individual PTFs by un-applying the appropriate PTFs.

Summary of Changes for VM/ESA Version 2 Release 3.0

Changes

- ***CMS Pipelines:*** All internal EXECs, messages, and modules of CMS Pipelines have been renamed from a DMS to an FPL prefix. Some new messages have been added and all messages numbers have changed, with many message texts changed also. Refer to *VM/ESA: Conversion Guide and Notebook* for cross-reference lists of the message numbers, and refer to *VM/ESA: System Messages and Codes* for more information about these messages.
- ***VM/ESA GUI:*** The VM/ESA GUI function is now part of the CMS component and will be serviced as part of CMS.
- ***New Production Step:*** The production step now includes information on how to move Java™ and NetRexx files into the Byte File System (BFS).

Summary of Changes for VM/ESA Version 2 Release 2.0

New Function

- ***VMFREPL:*** A new exec, VMFREPL, supports the local modification of replacement maintained parts.

Additional Changes

- The HELPINST segment has been split into the HELPSEG and INSTSEG segments.

Chapter 1. Servicing Your System

This book contains procedures you can use to install the following kinds of service to VM/ESA®, Virtual Machine/Enterprise Systems Architecture:

- Preventive service from the VM/ESA Recommended Service Upgrade (RSU)
- Corrective (COR) Service
- Local service
- Local modifications
- Reworking local service

Notes:

1. This book is designed to be used with the *VM/ESA: VMSES/E Introduction and Reference*.
2. *VM/ESA: VMSES/E Introduction and Reference* discusses the concepts of VM/ESA service and provides reference information about the VMSES/E service tools and files. If you need to become familiar with service using VMSES/E, you should read the introductory chapters in *VM/ESA: VMSES/E Introduction and Reference* ("Chapter 1. Introducing VMSES/E" and "Chapter 8. VM/ESA Service Concepts" would be recommended) before you begin using the procedures in this book.
3. Also, from time to time, this book points to reference information contained in *VM/ESA: Planning and Administration*.

   This book also refers to *VM/ESA: Installation Guide* for some procedures.

Overview of the Product Service Upgrade (PSU) Procedure

The PSU procedure is the method of installing preventive maintenance to VM/ESA Version 2 Release 4.0.

The Product Service Upgrade (PSU) is a procedure that refreshes the service database disks defined for VM/ESA Version 2 Release 4.0, leaves customized files untouched, and preserves any reach-ahead service. This procedure utilizes the VM/ESA Recommended Service Upgrade (RSU), which has all recommended service preinstalled (preapplied and prebuilt).

The PSU procedure cannot be used to migrate from one release to another, but is intended to upgrade your current VM/ESA Version 2 Release 4.0 service with the RSU tape. Because service from the RSU tape is preapplied and objects are already built, much time can be saved. The saved segments and nuclei must be rebuilt because they are not included on the tapes.

Overview of the Service Procedure

Install service to VM/ESA one component at a time, in the following order:

VMSES/E	Virtual Machine Serviceability Enhancements Staged/Extended
REXX/VM	REXX/VM
CMS	Conversational Monitor System

CP	Control Program
GCS	Group Control System
Dump Viewing Facility	Dump Viewing Facility
TSAF	Transparent Services Access Facility
AVS	APPC/VM VTAM® Support

When you service a VM/ESA component, you use a series of commands to accomplish the main service tasks:

- Merging service
- Receiving service for the component
- Applying service to the component
- Reworking local service (if applicable)
- Rebuilding the component
- Placing your serviced components into production.

The software inventories are updated when receiving, applying and building service.

The APPLY disks are called the alternate, intermediate, and production disks. They are used as follows:

- The **alternate** disk is used as a work disk, or staging area, during the installation of new service
- The **intermediate** disk contains the last level of service that was installed
- The **production** disk contains all of the previous levels of service that were installed.

You will see in greater detail how these disks are used in the following sections.

Figure 1-1 on page 1-3 shows the flow of information from disk to disk when you install service for a component.

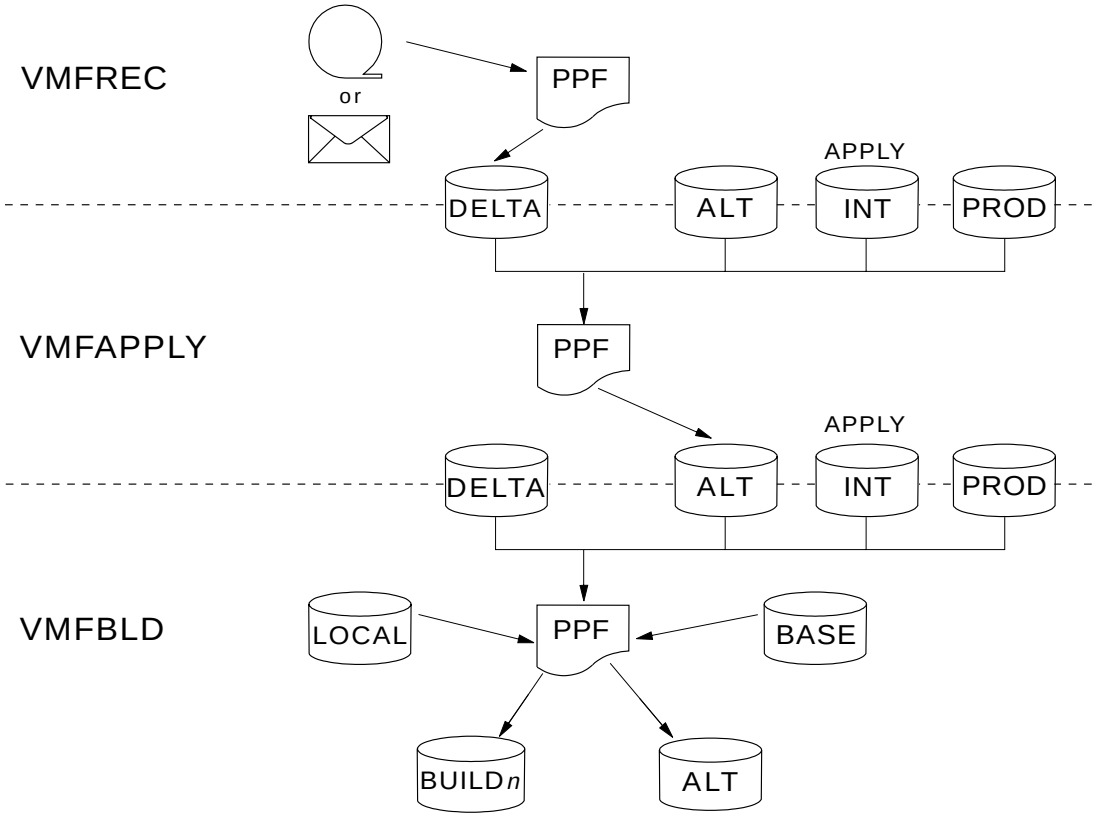

Figure 1-1. Disks Used When Installing Service

Merging Service

Use the VMFMRDSK command to clear the alternate apply disk (or intermediate disk, or both). The VMFMRDSK command merges files from one disk to another and then clears the disk that the code was merged from before receiving new service. This lets you remove the new service if a serious problem is found.

Also, the VMFMRDSK command could be used to merge all service to the production apply disks after the new service has been put into production. For more information about the VMFMRDSK command, see the *VM/ESA: VMSES/E Introduction and Reference*.

Receiving Service

The VMFREC command receives service from the delivery media and places it on the delta disk.

Note: If you are installing COR service, you can receive all of the service for a component (from multiple tapes) before applying and building that component.

Applying Service

The VMFAPPLY command updates the version vector table (VVT), which identifies the service level of all serviced parts. In addition, AUX files are generated from the VVT for parts which require them. VMFAPPLY only modifies the alternate apply disk. All lower level apply disks are unaffected.

Reworking Local Service (If Applicable)

All local service must be entered into the software inventory to allow VMSES/E to track the changes and build them into the system. It is also possible that IBM service can affect your current local service and you will need to rework your local service. **If a local change is not added to the software inventory, VMSES/E will not build it into the system.** See Chapter 5, “Local Service and Modifications — Overview” on page 5-1.

Building New Levels

The build task is accomplished by using the appropriate command or set of commands to generate a serviced level of an object on a build disk. When you run any build command, all local and service disks must be accessed to ensure that the highest level parts are available to generate an object. The generated object is placed on a build disk.

Placing the Serviced Components into Production

After you have tested the new service and are satisfied with the results for all serviced components, you should put them into production. You need to:

- Rebuild and save your Named Saved System (NSS) and Discontiguous Saved Segments (DCSS)
- Place the test build disks into production
- Rebuild the CMS saved system
- Place the CP nucleus and CMS into production
- Place TSAF, AVS and GCS into production
- Merge your tested service to your production service disks.

Service Dependencies among Components

Figure 1-2 on page 1-5 illustrates the build dependencies among the components.

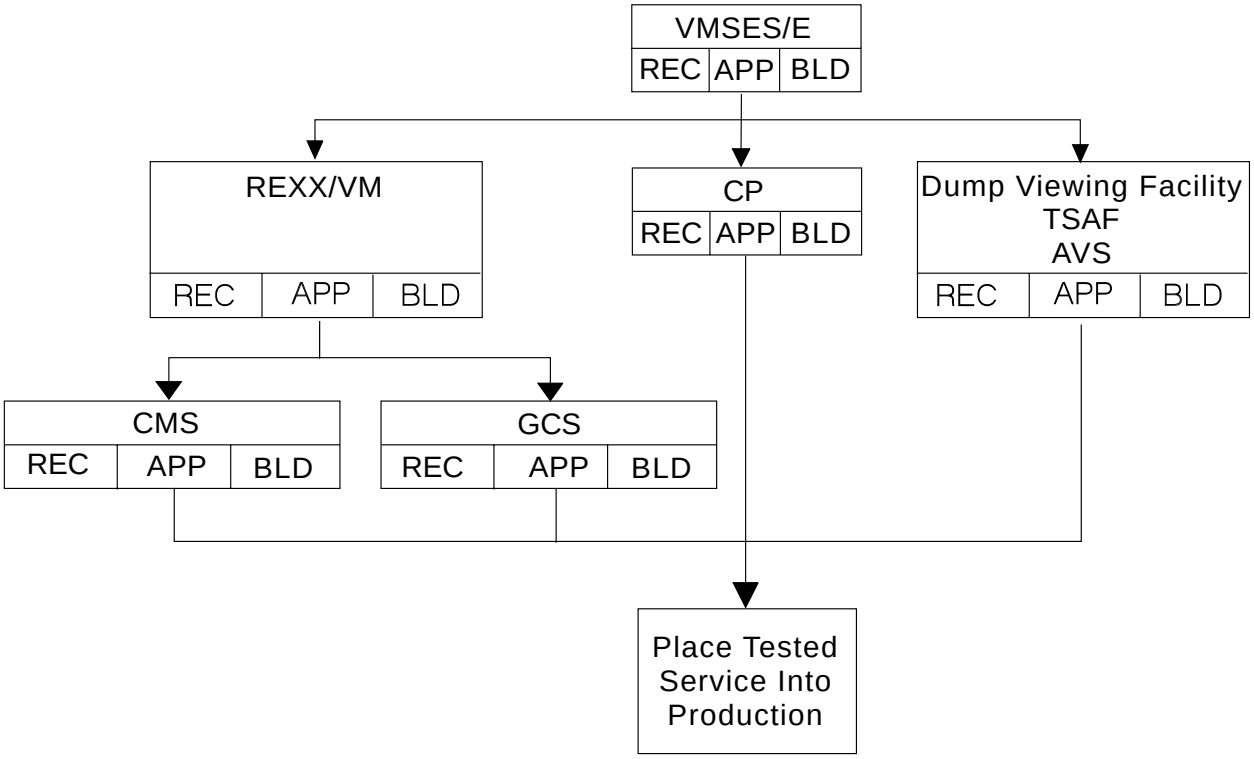

Figure 1-2. Component Dependencies for Installing Service

All components have a dependency on VMSES/E. It should be serviced first, so that the service tools are at the latest level when you service the rest of the components. In addition, because REXX/VM is part of the CMS and GCS nuclei, you must receive and apply REXX/VM service before you build CMS and GCS.

Recommended Order of Service

Because of these dependencies, IBM recommends that you service VM/ESA components in the following order:

1. VMSES/E—Receive, Apply, and Build

   The VMSES/E component is always serviced first. It contains the service tools that are required to install service for all VM/ESA components.

2. REXX/VM—Receive, Apply, and Build

   The build step for REXX/VM is a non-nucleus build. The serviced parts of REXX/VM are built into the CMS and GCS nuclei. You must receive and apply CMS and GCS service (if any) before you add REXX/VM service to the nuclei. This ensures that any requisites for these components are installed.

3. CMS—Receive, Apply, and Build

   After REXX/VM service has been received and applied, CMS service should be processed next. REXX/VM service must be received and applied before you build the CMS nucleus.

   When building CMS, it is important to note that the VM callable services libraries, SFS modules, REXX/VM, and CMS nucleus are all functionally

dependent on each other. For CMS to function correctly, these parts must be put into production at the same time.

4. CP—Receive, Apply, and Build
5. GCS—Receive, Apply, and Build

   After REXX/VM service has been received and applied, GCS service can be processed. REXX/VM service must be processed before you build the GCS nucleus, because the REXX/VM service is included in the GCS nucleus.

6. Dump Viewing Facility, TSAF, and AVS—Receive, Apply, and Build

   These components can be serviced in any order.

Out-of-Component Requisites

The service stream may introduce other dependencies not shown in Figure 1-2 on page 1-5. When a PTF requires service to several components, it introduces one or more out-of-component requisites. VMFAPPLY identifies any out-of-component requisites in the apply message log.

If you find any out-of-component requisites in the apply message log, you must receive, apply, and build all affected components before you put any components into production.

Service Packaging

VM/ESA service is packaged on its delivery media at the component level. Each component has a separate source product parameter file ($PPF) that identifies the service for the component on the delivery media.

In addition to the base $PPFs for each component, an override $PPF file (ESA $PPF) is provided for VM/ESA that identifies the component names. This override file makes it easier to enter the commands for each component, because you can use the component name rather than the base $PPF's file name.

Table 1-1 on page 1-7 lists the file names of the base $PPFs, descriptions of each file, and their component names as defined in the ESA $PPF file. Samples of these $PPFs are listed in the *Installation and Service Sample Files* informal document packaged with the VM/ESA product.

Table 1-1. VM/ESA Component PPF File Names and Aliases

Base $PPF File Name	Description	Component Name as Defined in ESA $PPF
2VMVMK40	VMSES/E Product Parameter File	VMSES
2VMVMF40	REXX/VM Product Parameter File	REXX
2VMVMA40	CMS Product Parameter File	CMS
2VMVMB40	CP Product Parameter File	CP
2VMVML40	GCS Product Parameter File	GCS|GCSSFS
2VMVMI40	Dump Viewing Facility Product Parameter File	DV
2VMVMH40	TSAF Product Parameter File	TSAF|TSAFSFS
2VMVMD40	AVS Product Parameter File	AVS|AVSSFS

Table 1-2 lists other $PPF files that are provided by VM/ESA.

Table 1-2. NLS $PPF Files

$PPF File Name	Description
FRANC	French Product Parameter File
CANFR	Canadian French Product Parameter File
GER	German Product Parameter File
KANJI	Japanese Product Parameter File
UCENG	Uppercase English Product Parameter File

When servicing a national language (including uppercase English), use the $PPF name in Table 1-2 during part of the build service task (for example, building a nucleus). For the receive service task and apply service task you can use the ESA $PPF file. If you need to make updates to the system, you should use the appropriate NLS $PPF as the override instead of the ESA $PPF.

Disks Used to Service VM/ESA

The following sections describe how the service disks are used during the installation of service.

Each component uses a set of disks to control the installation of service and local modifications. CP and the Dump Viewing Facility share the same set of disks. CMS and REXX/VM share the same set of disks. AVS and TSAF share the same set of disks (though they have unique SFS directories). VMSES/E and GCS each have a unique set of disks or SFS directories.

A number of service disks are defined and used as follows:

TASK The task disks contain any files that you want accessed before the service disks defined for a component. No default task disks are defined in VM/ESA Version 2 Release 4.0.

LOCALMOD The local disk (*n*C4) contains any local modifications you have installed, except PTF fixes.

LOCALSAM The local sample disk (*n*C2) contains the sample and example files you receive from IBM, for example, SYSTEM CONFIG, LOGO CONFIG, HCPRIO, HCPSYS, and DMSNGP.

DELTA The delta disk (*n*D2) contains all files received from a service tape, plus the software inventory files generated by the VMFREC command. Files on this disk include PTF-numbered parts, updates, and PTF information (such as PTFs received, requisites to those PTFs, and APAR descriptions). There is one default disk defined as a delta disk for each component in VM/ESA.

APPLY The apply disks identify the service level of a component, and the service level of each part. These disks contain the software inventory files used by the VMFAPPLY and VMFBLD commands. This includes AUX files. There are three default disks defined as apply disks for each component in VM/ESA: the alternate apply disk, the intermediate apply disk, and the production apply disk.

- **Alternate Apply disk (*n*A6)** This disk is used as a staging disk for VMFAPPLY and VMFBLD processing. This disk is only used as a staging disk to allow for recovery if VMFAPPLY or VMFBLD fails.
- **Intermediate Apply disk (*n*A4)** This disk contains the level of service that was last installed. The purpose of the intermediate disk is to separate service levels so that you can easily back out the level of service on the alternate disk. This disk is also used as the test level of the system.
- **Production Apply disk (*n*A2)** This disk contains the level of service that has been placed into production. When a component is placed into production all files that reside on the intermediate apply disk are moved to the production apply disk (using VMFMRDSK).

BUILD*n* The build disks contain all the usable forms (objects) built by VMFBLD.

BASE*n* The base disks contain the object code and base source code for a component. The base disks are never updated during the service process.

SYSTEM The system disks are disks you want accessed after the service disks defined for a component.

The following charts represent the disks used to service the components of VM/ESA. Note that some of the components share the same disks for service, such as CP and the Dump Viewing Facility.

VMSES/E Minidisks

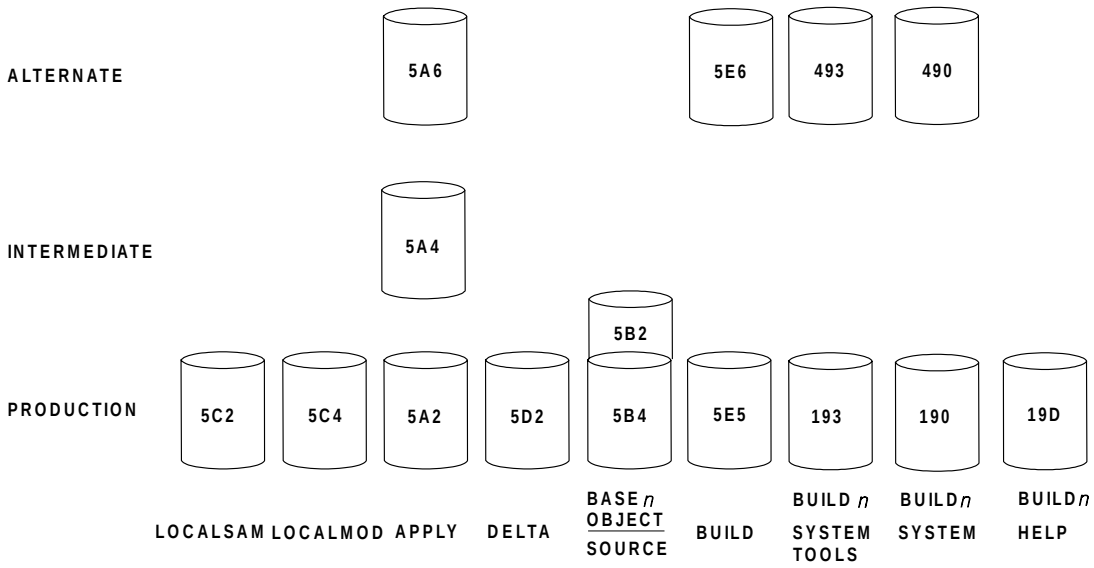

Figure 1-3. Disks Used When Servicing VMSES/E

CMS and REXX/VM Minidisks

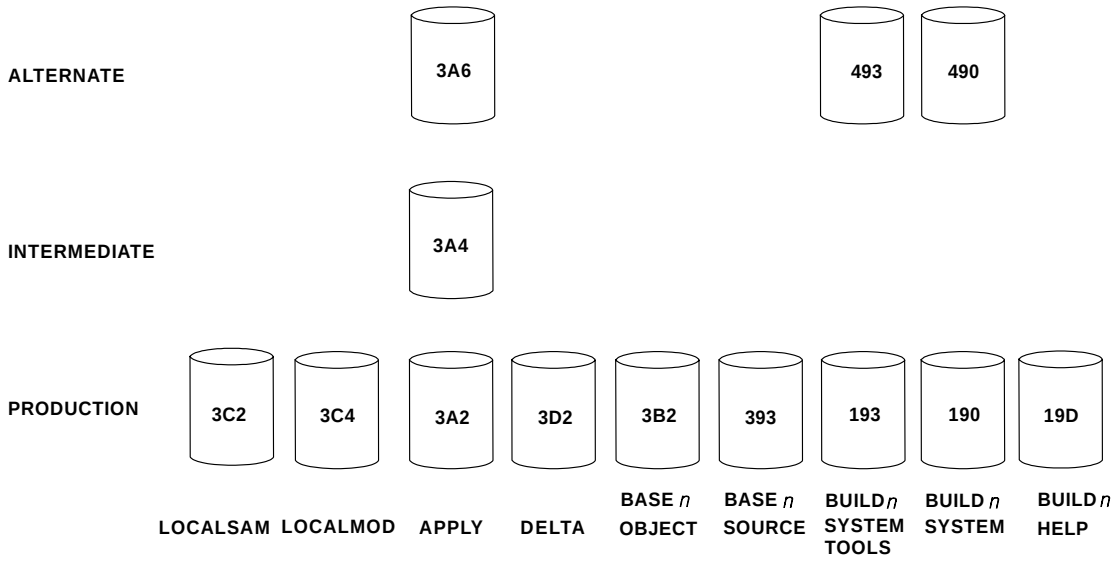

Figure 1-4. Disks Used When Servicing CMS and REXX/VM

CP and Dump Viewing Facility Minidisks

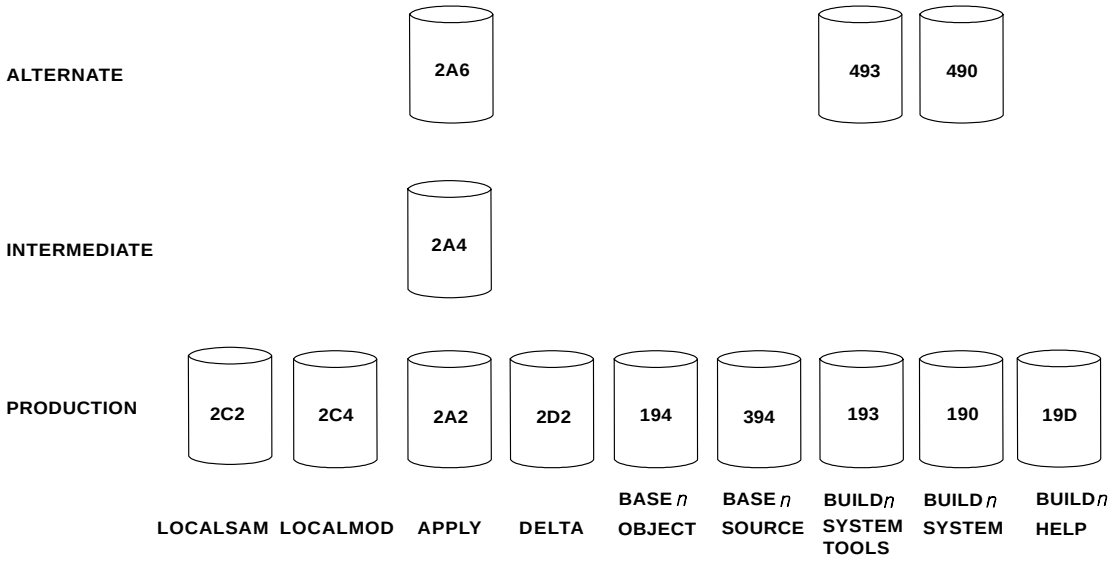

Figure 1-5. Disks Used When Servicing CP and Dump Viewing Facility

GCS Minidisks

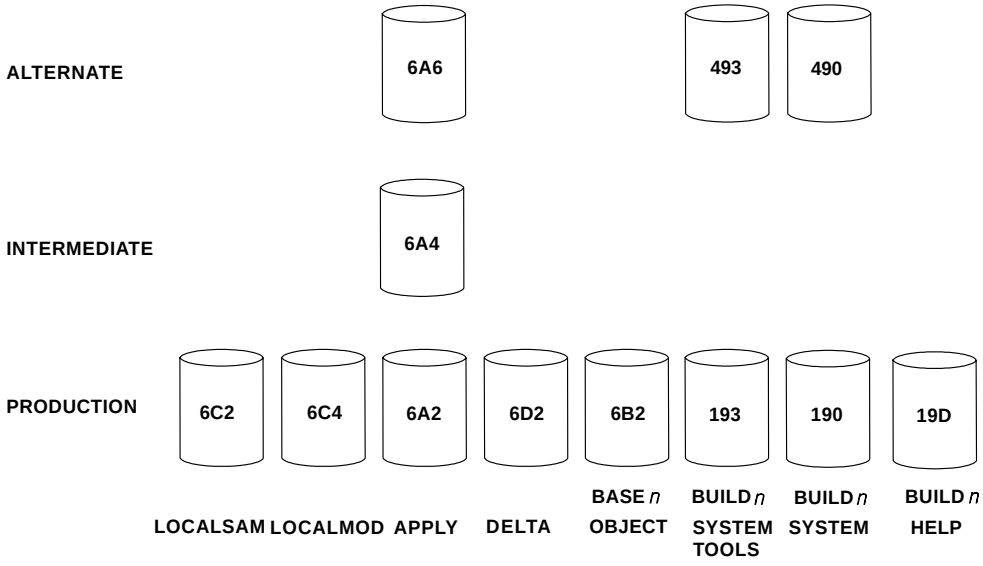

Figure 1-6. Disks Used When Servicing GCS

Note: If you are installing GCS to the Shared File System (SFS), then the following directory IDs match the minidisk definitions.

GCS Directory IDs

Table 1-3. GCS Minidisks and comparable Directory IDs

Minidisk	Directory ID
6A2	VMSYS:MAINT.GCS.APPLYPROD
6A4	VMSYS:MAINT.GCS.APPLYINT
6A6	VMSYS:MAINT.GCS.APPLYALT
6D2	VMSYS:MAINT.GCS.DELTAPROD
6B2	VMSYS:MAINT.GCS.OBJECT

TSAF and AVS Minidisks

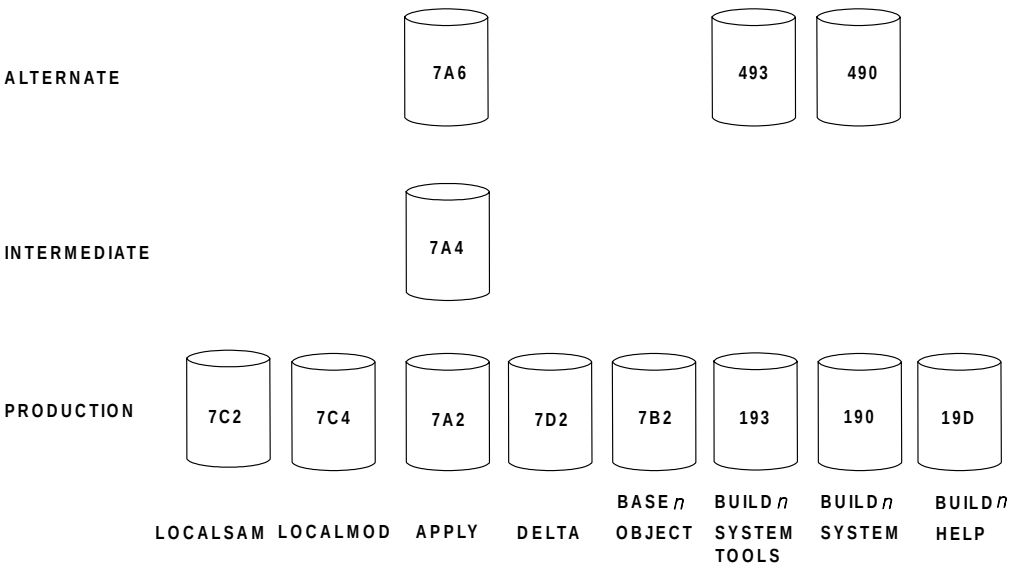

Figure 1-7. Disks Used When Servicing TSAF and AVS

Note: If you are installing TSAF to the Shared File System (SFS), then the following directory IDs match the minidisk definitions.

TSAF Directory IDs

Table 1-4. TSAF Minidisks and comparable Directory IDs

Minidisk	Directory ID
7A2	VMSYS:MAINT.TSAF.APPLYPROD
7A4	VMSYS:MAINT.TSAF.APPLYINT
7A6	VMSYS:MAINT.TSAF.APPLYALT
7D2	VMSYS:MAINT.TSAF.DELTAPROD
7B2	VMSYS:MAINT.TSAF.OBJECT

Note: If you are installing AVS to the Shared File System (SFS), then the following directory IDs match the minidisk definitions.

AVS Directory IDs

Table 1-5. AVS Minidisks and comparable Directory IDs

Minidisk	Directory ID
7A2	VMSYS:MAINT.AVS.APPLYPROD
7A4	VMSYS:MAINT.AVS.APPLYINT
7A6	VMSYS:MAINT.AVS.APPLYALT
7D2	VMSYS:MAINT.AVS.DELTAPROD
7B2	VMSYS:MAINT.AVS.OBJECT

Building New Objects

There are three basic build steps that you need to perform after you apply service:

- Enter VMFBLD with the STATUS option to identify all objects that need to be built, because of the application of service.
- Enter VMFBLD with the SERVICED option to generate executable objects that need to be built, because of the application of service.
- Perform post-build tasks for objects that cannot be generated by VMFBLD.

VM/ESA provides several functions that support these basic build steps for different parts of the system. The function that is required depends on the type of object being generated and the environment in which it runs.

Build Tasks by Component

Table 1-6 identifies each major build task and the service EXEC, CMS command, or utility program you use to accomplish the task for each component.

NA (not applicable) indicates that a task does not apply to the component.

Table 1-6 (Page 1 of 2). Build Tasks for VM/ESA Components

Build Task	VMSES/E	REXX/VM	CMS	CP	GCS	Dump Viewing Facility	TSAF	AVS
Update $Source Files	VMFEXUPD	VMFEXUPD	VMFEXUPD	VMFEXUPD	NA	VMFEXUPD	NA	NA
Rebuild MACLIBs	NA	VMFBLD	VMFBLD	VMFBLD	VMFBLD	NA	NA	NA
Reassemble Source Files	NA	VMFHLASM	VMFHLASM	VMFHLASM	NA	NA	NA	NA
Rebuild NLS Parts	VMFNLS	NA	VMFNLS	VMFNLS	VMFNLS	VMFNLS	VMFNLS	VMFNLS
Select and Copy Serviced Files	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD
Rebuild TXTLIBs	NA	NA	VMFBLD	NA	NA	NA	NA	NA
Rebuild Modules	VMFBLD	NA	VMFBLD	VMFBLD	NA	VMFBLD	VMFBLD	NA
Rebuild LOADLIBs	NA	NA	VMFBLD	NA	VMFBLD	NA	NA	VMFBLD
Rebuild CSLLIBs	NA	NA	VMFBLD	NA	NA	NA	NA	NA

Table 1-6 (Page 2 of 2). Build Tasks for VM/ESA Components

Build Task	VMSES/E	REXX/VM	CMS	CP	GCS	Dump Viewing Facility	TSAF	AVS
Rebuild the F Assembler	NA	NA	VMFBLD	NA	NA	NA	NA	NA
Rebuild Nuclei	NA	NA	VMFBLD	VMFBLD	VMFBLD	NA	NA	NA
Rebuild Saved Segments	NA	NA	VMFBLD, others	NA	NA	NA	NA	NA

Note: Update $Source Files, Reassemble Source Files and Rebuild NLS Parts are optional procedures.

Using Tools for Service and System Generation

VM/ESA provides a number of tools to help you perform install, service, and system generation tasks. Table 1-7 lists VM/ESA install, service, and system generation EXECs and the books describing each EXEC. Use the following key for this table.

Abbreviation	Title
VMSES/E I and R	*VM/ESA: VMSES/E Introduction and Reference*
Install	*VM/ESA: Installation Guide*
CMS Cmd Ref	*VM/ESA: CMS Command Reference*
CP Cmd Ref	*VM/ESA: CP Command and Utility Reference*
GCS Ref	*VM/ESA: Group Control System*

Table 1-7 (Page 1 of 4). VM/ESA Install, Service, and System Generation Tools

Tool	Task	Book
ASSEMBLE	Processes source statements in assembler language source files.	CMS Cmd Ref
CSLGEN	Builds a callable services library (CSL).	CMS Cmd Ref
DCSSGEN	Builds the CMS installation saved segment (CMSINST).	CMS Cmd Ref
DIRECTXA	Creates a user directory.	CP Cmd Ref
DIRONLIN	Brings the directory online that was built by INSTDIR.	Install
DISKMAP	Summarizes the MDISK statements in the user directory. The output shows gaps and overlaps between minidisk assignments.	CP Cmd Ref
DOSGEN	Builds the CMSDOS physical saved segment.	CMS Cmd Ref
EXECUPDT	Produces an updated version of a $Source file.	CMS Cmd Ref
EXPAND	Adds space to a program in object deck form.	VMSES/E I and R
GENCPBLS	Updates the CP load list build list.	VMSES/E I and R
GENMOD	Generates CMS module files.	CMS Cmd Ref
GROUP	Builds a GCS configuration file.	GCS Ref
HCPLDR	Calls and controls the system loader.	CP Cmd Ref
INSTALL	Loads base and optional components to disks.	Install
INSTDEF	Tailors CP, CMS, and GCS.	Install
INSTDEF2	Rebuilds segments and updates USER DIRECT file.	Install

Table 1-7 (Page 2 of 4). VM/ESA Install, Service, and System Generation Tools

Tool	Task	Book
INSTDIR	Builds a directory for your installation.	Install
INSTFPP	Installs optional products.	VMSES/E I and R
INSTIIS	Formats and labels your installation DASD and restores the IIS.	Install
INSTPLAN	Selects items to load and DASD type on which to install.	Install
INSTPOOL	Starts the file pool servers during installation procedures.	Install
ITNVTSTR	Processes install and service orders delivered by Advanced Digital Delivery or Service Update Facility/390.	VMSES/E I and R
LANGGEN	Loads national language text files into a saved segment.	CMS Cmd Ref
LANGMERG	Combines national language files for an application into a single text file.	CMS Cmd Ref
LOADLIB	Lists, copies, or compresses CMS load libraries.	CMS Cmd Ref
MIGR51D	Migrates and updates the System Software Inventory files.	Install
MOVE2SFS	Moves data for GCS, TSAF, and AVS from minidisks to Shared File System (SFS) servers.	Install
POSTDDR	Creates software inventory tables.	Install
POSTLOAD	Performs cleanup tasks depending on what you have loaded.	Install
PRELOAD	Collects multiple text files and reformats them into a single text file.	CMS Cmd Ref
SAMGEN	Builds the CMSBAM physical saved segment.	CMS Cmd Ref
SAMPNSS	Defines named saved systems.	CMS Cmd Ref
SAVEFD	Places file directory information for a shared, extended data format (EDF) R/O minidisk into a discontiguous shared segment (DCSS).	CMS Cmd Ref
SEGGEN	Builds logical saved segments defined in a physical saved segment.	CMS Cmd Ref
SNTINFO	Gets discontiguous saved segment (DCSS) information directly from CP.	VMSES/E I and R
SPXTAPE	Saves standard spool files and system data files on tape and restores SPXTAPE-format files from tape to the spooling system.	CP Cms Ref
UTILITY	Provides occasionally-used installation functions, such as, issuing DIAGNOSE code X'24' and X'210' for a virtual device and creating a stand-alone service utility tape for either or both ICKDSF and DDRXA.	CP Cmd Ref
VMFAPPLY	Updates the maintenance level of the specified product.	VMSES/E I and R
VMFASM	Updates an ASSEMBLE source file according to entries in a control file, then assembles the source file to produce an object file.	VMSES/E I and R
VMFBLD	Builds objects for the specified product.	VMSES/E I and R
VMFCNVT	Converts size and block size data into cylinders and displays the results.	VMSES/E I and R
VMFCOPY	Copies a file to a VMSES/E target minidisk or SFS directory and updates the parts catalog table on that target.	VMSES/E I and R
VMFERASE	Erases a file on a VMSES/E target minidisk or SFS directory and updates the parts catalog table on that target.	VMSES/E I and R
VMFEXUPD	Calls the EXECUPDT command to apply updates to a $Source program.	VMSES/E I and R

Table 1-7 (Page 3 of 4). VM/ESA Install, Service, and System Generation Tools

Tool	Task	Book
VMFHASM	Updates an ASSEMBLE source file according to entries in a control file, then uses the H assembler to produce an object file.	VMSES/E I and R
VMFHLASM	Updates an ASSEMBLE source file according to entries in a control file, then uses the HL assembler to produce an object file.	VMSES/E I and R
VMFINFO	Queries the Software Inventory tables.	VMSES/E I and R
VMFINS	Installs, migrates, builds, and deletes products.	VMSES/E I and R
VMFLKED	Link edits modules into a load library (LOADLIB).	CMS Cmd Ref
VMFMAC	Builds macro libraries (MACLIBs) containing macro and copy files.	CMS Cmd Ref
VMFMERGE	Applies PTFs to Systems Network Architecture (SNA) products. VMFMERGE is used only to service SNA products.	VMSES/E I and R
VMFMRDSK	Consolidates the contents of minidisks/directories within a string.	VMSES/E I and R
VMFNLS	Applies updates to national language files and compiles the updated versions.	VMSES/E I and R
VMFOVER	Creates a temporary PPF by applying overrides to a source PPF.	VMSES/E I and R
VMFPLC	Provides a front end to routines that use VMFPLC2 when conversion to VMFPLCD or a dual path is desired.	CMS Cmd Ref
VMFPLCD	Loads files from an envelope, dumps files to an envelope, and controls various envelope operations.	CMS Cmd Ref
VMFPLC2	Loads files from tape, dumps files to tape, and controls various tape drive operations.	CMS Cmd Ref
VMFPPF	Compiles a source PPF into its usable form.	VMSES/E I and R
VMFPSU	Helps you choose which method to use when you install a Product Service Upgrade (PSU).	VMSES/E I and R
VMFQMDA	Displays the current VMSES/E access order.	VMSES/E I and R
VMFQOBJ	Returns information about objects defined in build lists.	VMSES/E I and R
VMFREC	Processes installation and service tapes.	VMSES/E I and R
VMFREPL	Supports the local modification of replacement maintained parts.	VMSES/E I and R
VMFREM	Removes service.	VMSES/E I and R
VMFREMOV	Removes PTFs from Systems Network Architecture (SNA) products. VMFREMOV is used only to service SNA products.	VMSES/E I and R
VMFSETUP	Sets up a minidisk and SFS directory access order, or detaches minidisks that were linked by previous invocations of the VMFSETUP EXEC, depending on how it is invoked.	VMSES/E I and R
VMFSGMAP	Processes and displays the saved segment information defined in a saved segment configuration build list and save segment data file.	VMSES/E I and R
VMFSIM	Provides an interface to the Software Inventories.	VMSES/E I and R
VMFSUFIN	Installs RSU and/or COR service from envelope files in support of Service Update Facility/390.	VMSES/E I and R
VMFSUFTB	Builds the VM SYSSUF table for use by Service Update Facility/390.	VMSES/E I and R
VMFTXT	Builds a text library (TXTLIB) from text decks.	CMS Cmd Ref
VMFVIEW	Displays message logs using XEDIT with predefined PF keys.	VMSES/E I and R
VMFZAP	Applies ZAPs to Systems Network Architecture (SNA) products. VMFZAP is used only to service SNA products.	VMSES/E I and R

Table 1-7 (Page 4 of 4). VM/ESA Install, Service, and System Generation Tools		
Tool	**Task**	**Book**
ZAP	Modifies or dumps MODULE, LOADLIB, or TXTLIB files.	CMS Cmd Ref
ZAPTEXT	Modifies or dumps individual text files.	VMSES/E I and R

Notes:

[1] EXPAND is called by and described under ZAPTEXT.

[2] VMFMERGE, VMFREMOV, and VMFZAP are used **only** to service Systems Network Architecture (SNA) products.

Using the Online HELP Facility

You can receive online information about VMSES/E commands using the VM/ESA HELP Facility. For example, to display a menu of VMSES/E commands, enter:

```
help vmses menu
```

To display information about a specific VMSES/E command (VMFREC in this example), enter:

```
help vmses vmfrec
```

For more information about using the HELP Facility, see the *VM/ESA: CMS User's Guide*. To display the main HELP Task Menu, enter:

```
help
```

For more information about the HELP command, see the *VM/ESA: CMS Command Reference* or enter:

```
help cms help
```

Chapter 2. Using the Product Service Upgrade (PSU) Procedure

You can upgrade your existing VM/ESA Version 2 Release 4.0 system using the service files from the IBM VM/ESA Recommended Service Upgrade. **This is the method of installing preventive service to VM/ESA** that was introduced in VM/ESA Release 1.1. The procedure for performing a Product Service Upgrade from the VM/ESA RSU is outlined in this chapter.

Note: This procedure is not intended for use by customers migrating from a previous release of VM. It should **only** be used to upgrade from a previous service level of VM/ESA Version 2 Release 4.0.

The RSU that is processed using the PSU procedure has the following logical structure for each component.

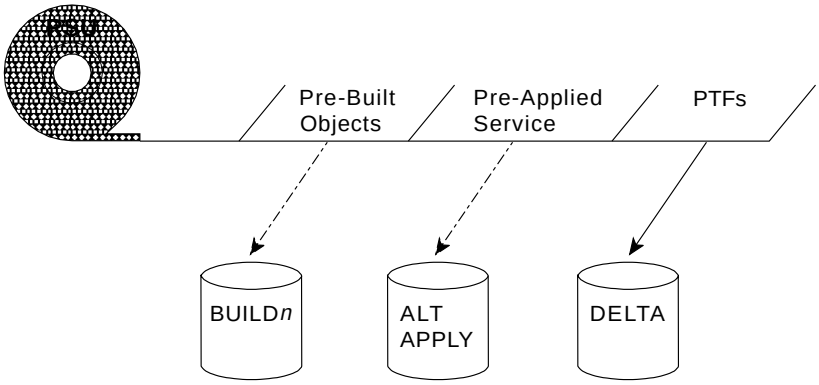

Figure 2-1. RSU and Files

The RSU is structured to allow you to use one of two methods to install the service for each component.

- The first method is to install all PTFs for a component included on the RSU plus the files containing the preapplied service and prebuilt objects. All PTF-related files are loaded to the delta disk. The file containing the preapplied service is loaded to the alternate apply disk and the contents of the files containing prebuilt objects are loaded to the appropriate build disks.

  This method saves considerable time if you have only a small number of PTFs already installed that are not contained on the RSU and do not want to exclude any PTFs on the RSU.

- The second method is to install all of the PTFs for a component included on the RSU and bypass the files containing the preapplied service and prebuilt objects. All PTF-related files are loaded to the delta disk. The file containing the preapplied service is bypassed and the contents of the files containing prebuilt objects are also bypassed. This results in having all PTFs received the same as if they were shipped on a COR tape. You must apply the PTFs received using VMFAPPLY and build all serviced objects using VMFBLD.

  This method avoids reapplying any PTFs that you have already installed and building objects that are affected, and allows you to exclude PTFs that you do

not want, which are included on the RSU. It also avoids altering any objects that you have already built on the test build disks.

Points to consider about using the Product Service Upgrade procedure are:

- This process will not alter any of your tailored files in any way
- Planning must be done (such as determining disk sizes, and determining what service, if any, on your existing system is not contained on the RSU) prior to actually loading the service from the RSU. These tasks will be discussed.
- CMS, CP and GCS nuclei, as well as saved segments need to be rebuilt to include the new service.

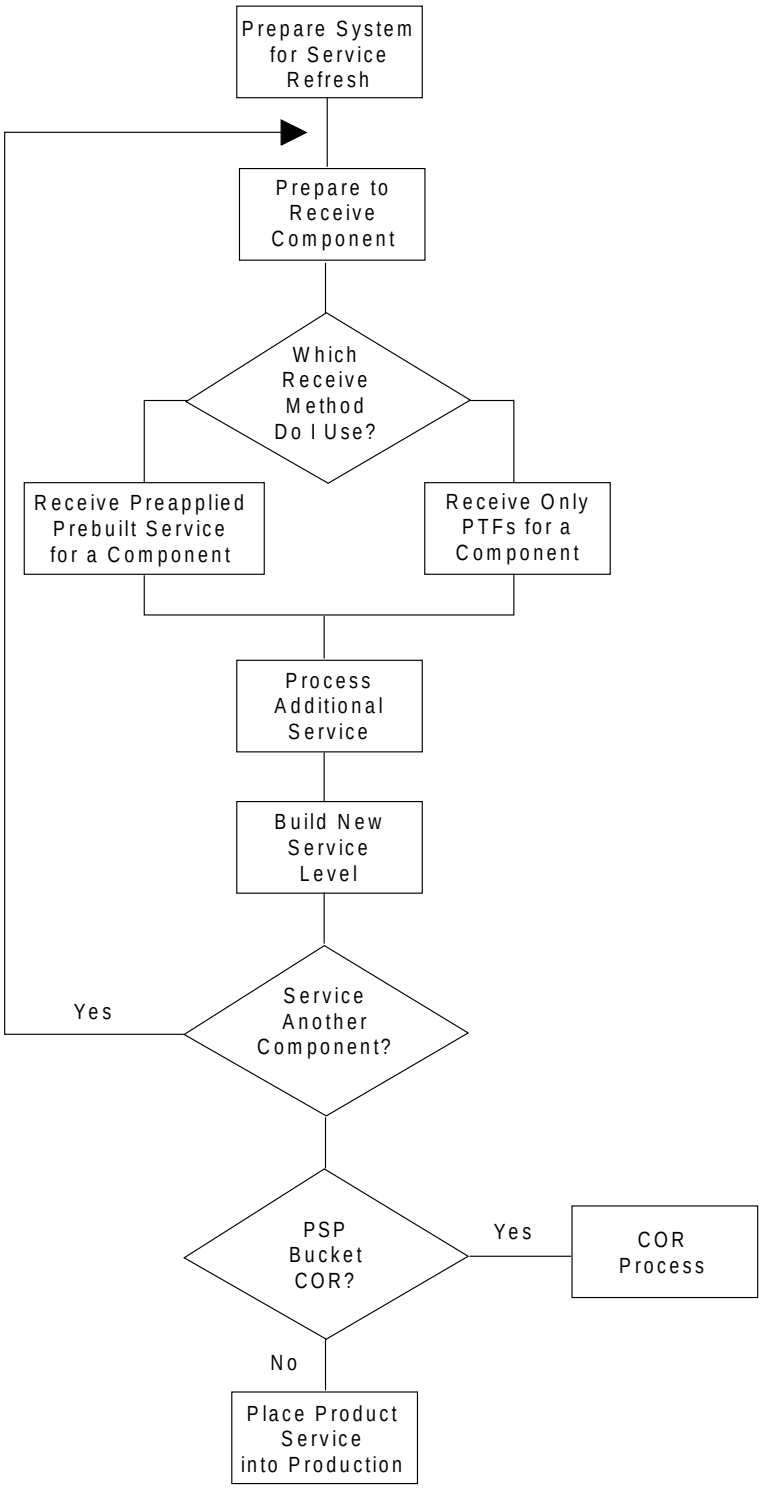

Figure 2-2. Service Application Flowchart using PSU

The following outline is an overview of what tasks need to be performed during the PSU procedure.

1. Prepare Your System for Service Refresh

"Prepare Your System for Service Refresh" on page 2-5 helps you to prepare your system to receive the service from the RSU.

In this task you will receive the documentation contained on the RSU and determine the DASD required to install the RSU. You will also receive the files used to process the RSU.

The following steps (2 - 6) should be performed for each component in VM/ESA, using the order of application of service described in Chapter 1, Servicing Your System.

2. Prepare to Receive a Component

   In the task, "Prepare to Receive a Component from the RSU" on page 2-9, you will set up your environment.

3. Which Receive Method Do I Use?

   Invoke the VMFPSU command. This creates an output file which lists the number of new PTFs, PTFs currently applied that are not on the RSU and PTFs to be excluded. The command also identifies local modifications that are affected by RSU service.

   If possible, you should follow the process to "Receive Preapplied, Prebuilt Service" (Preferred Method). This is the preferred method. However, if you have a large number of PTFs to reapply or you want to exclude PTFs from the RSU then you need to follow the procedure to "Process Only the PTFs from the RSU" (Alternate Method).

   - Process Preapplied, Prebuilt Service
     - Preferred Method: Receive Preapplied, Prebuilt Service

       page 2-11 contains the receive process for the RSU.
   - Process Only PTFs from the RSU
     - Alternate Method: Receive Only PTFs from RSU

       page 2-11 contains the process to receive only the PTFs contained on the RSU. This procedure installs the service as if it was corrective service.

4. Process Additional Service for a Component

   "Process Additional Service" on page 2-12 will reapply any reach-ahead service that may be on the system or indicate that there are no reach-ahead PTFs to apply. If necessary, you might also need to rework local service.

5. Rebuild Component

   After you have received the PTFs, you may need to apply the service and rebuild the component using the COR procedures defined for the component in Chapter 3, Using the COR Service Procedure.

   Finally, "Build the New Service Level" on page 2-14 identifies the build sections required for each component. After completing the build steps for the component, you should continue with the next component to be processed.

6. Service Another Component?

   If you have another component to service, repeat the steps in this outline, beginning with step 2.

7. Process PSP Bucket COR Tape

If you received the Preventive Service Planning (PSP) bucket COR tape with your service, you should process it now. For the component(s) that have service, follow the individual service steps (in Chapter 3, “Using the COR Service Procedure” on page 3-1). You will not return to this procedure.

8. Place Your Serviced Components into Production

   In this step you will place the new service into production. This is the last step in the PSU Procedure.

Materials Needed for a Product Service Upgrade

- Obtain the latest VM/ESA RSU:

  You need to order the VM/ESA RSU. You can order the RSU through the IBM Support Center.

- In addition to this book, you will also need the following documentation:
  - *VM/ESA: Installation Guide* is supplied with your initial product order.
  - *VM/ESA: VMSES/E Introduction and Reference*
- Also, contact your support center for the latest installation information, for example, the RSU PSP bucket.

Prepare Your System for Service Refresh

1 Log on MAINT.

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

2 Access the required disks for the service procedure, if necessary.

access *vdev* **a**	*vdev* is the virtual address of your R/W A-disk.
access 5e5 b	Access the VMSES/E installation and service tools disk.
access 51d d	Access the software inventory disk R/W.

3 Setup for the RSU media type (tape or envelope).

If tape:

attach *rdev* *** 181**	*rdev* is the real address of the tape drive containing the RSU which must also be attached as 181.

If envelope:

Follow the procedures in the *ServiceLink User's Guide* or the appropriate README file to retrieve and decompact the envelope file on your A-disk. The *filename* of the decompacted file will be of the format RPTF*num*. The *filetype* will be SERVLINK. You will need to enter the *filename* on VMFINS commands later in this procedure.

4 Receive the documentation and identify the components on the RSU. This step will also load the cumulative Version Vector Table (VVT) for each component, (*appid* VVTPSU), which identifies all service contained on the RSU for that component. These files are loaded to the 51D disk.

If tape:

vmfins install info (nomemo	NOMEMO will not print the memos.

If envelope:

vmfins install info (nomemo env rptf*num*	Use ENV rptf*num* if the RSU was received as an envelope file.

5 Review the following sections (at a minimum) in each RSU memo (*prodid* MEMO D).

- DASD Requirements
- Special Instructions
- Required Out of Component Requisites
- PTFs in Error.

Also review the hard copy memo that is shipped with the RSU tape. This memo has the latest information about the RSU schedules and also lists all the VM Licensed Products that have service on the VM/ESA RSU or ESO (Extended Service Option).

6 Determine DASD sizes for disks to receive service:

In order to receive the service from the RSU, you need to have adequate space available on each component's alternate APPLY and DELTA disks. The required sizes are identified in the product documentation (*prodid* MEMO D) received in the previous step.

Table 2-1. Component Name and Product Identifier

compname	*prodid* **and default** *appid* **Identifier**	**DELTA/APPLY Disks or Directories**	**Required 4K Blocks (see** *prodid* **MEMO)**
VMSES	2VMVMK40	5D2/5A6	
REXX	2VMVMF40	3D2/3A6	
CMS	2VMVMA40	3D2/3A6	
CP	2VMVMB40	2D2/2A6	
GCS	2VMVML40	6D2/6A6	
GCSSFS	2VMVML40	.gcs.deltaprod/ .gcs.applyalt	
DV	2VMVMI40	2D2/2A6	
TSAF	2VMVMH40	7D2/7A6	
TSAFSFS	2VMVMH40	.tsaf.deltaprod/ .tsaf.applyalt	
AVS	2VMVMD40	7D2/7A6	
AVSSFS	2VMVMD40	.avs.deltaprod/ .avs.applyalt	

Note: For GCS, TSAF, and AVS, use the *compname* SFS product identifier only if these components have been loaded in to shared file directories.

7 Prepare the system for service:

You need to prepare the system for service each time you apply new service to your system.

a. As a precaution, do a complete backup of your system. See the *VM/ESA: Installation Guide* for instructions on storing a backup copy of the VM/ESA system to tape.

b. Set up the correct virtual machine environment and IPL.

set machine xa
System reset.
System = XA

If your machine environment is already in the correct mode, you will not see all of these messages.

define storage *nn***M**
STORAGE CLEARED - SYSTEM RESET
STORAGE = nnM

Define the storage to at least 35MB. If you will be servicing GCS, make sure to define your storage to be at least as large as the value you defined for HIGH COMMON END storage (default 19MB) during the installation procedures. For more information, see *VM/ESA: Installation Guide*.

Defining storage causes a system reset.

ipl 190 clear

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

ENTER

Press ENTER to complete the IPL.

c. Access the required disks for the service procedure, if necessary.

access *vdev* **a**	You must have a R/W A-disk accessed.
access 5e5 b	Access the VMSES/E installation and service tools disk.
access 51d d	Access the software inventory disk R/W.

d. If you want to change the IBM service defaults, change the PPFs using $PPF overrides on the VMSES/E system inventory disk (51D) to reflect your system configuration. For information about tailoring the PPF, see *VM/ESA: VMSES/E Introduction and Reference*.

The variable *ppfname* is the name of your PPF override file. The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF. If you have your own PPF override file, you should use your file's *ppfname* throughout each procedure.

Note: If you have updated the 190 or 193 disks you should copy any new and changed files to the 490 and 493 disks. This will avoid accidentally back-leveling modified files.

8 Merge the APPLY disks for all components that you are going to service:

Next, you must prepare your system to receive the service from the RSU. To do this, you must first clear the alternate apply disk for receipt of the service from the RSU.

The merge for **all components** that you are going to service must be performed before continuing to the next task.

Enter the VMFMRDSK command to merge the alternate apply disk to the intermediate apply disk. This will clear the alternate apply disk. If components share the APPLY disk or directory, the merge only needs to be performed once for these components. For example, CMS and REXX use the 3A6 as the APPLY. These two components share these disks, so the following command only needs to be entered with either REXX or CMS as the *compname*.

vmfmrdsk esa *compname* **apply (setup**	*compname* is the component name. See Table 2-1 on page 2-7 for the values unless you are using an override *compname*.

Note

The following tasks, Prepare to Receive a Component from the RSU, Which Receive Method Do I Use?, Receive a Component from the RSU, Process Additional Service, and Build the New Service Level should be completed for each component contained on the RSU, using the order of application of service described in Chapter 1. Perform these tasks for each component. Then, if you have service for another component, repeat these tasks. After all components have been handled, *then and only then* can you test and place the components into production.

Prepare to Receive a Component from the RSU

1 Generate the appropriate level of PPF. You only need to do this if you modified your PPF override file.

vmfppf *ppfname compname*

ppfname here is the file name of your override PPF file. *compname* here is the override component name in your override PPF file.

2 Set up your access order.

vmfsetup esa *compname*

Which Receive Method Do I Use?

Before receiving the service you must decide which method is best for you. Use the VMFPSU command to help make this decision.

Preferred Method: “Receive the Preapplied, Prebuilt Service”

will receive the PTFs along with the preapplied service and prebuilt objects. This is the **preferred method**. This method, however, **does not** allow you to exclude PTFs. You will also need to reapply any reach ahead PTFs and build all affected objects on your system.

Alternate Method: “Receiving PTFs Only from the RSU Tape”

will receive the PTFs as if they were corrective service. If you have a large number of reach ahead PTFs that would have to be reapplied if you used the preapplied, prebuilt service, you might want to use this method. If you have PTFs that you **want to exclude** from the RSU, then you **must** use this method. You will have to apply all new PTFs on the RSU and build all affected objects.

1 Enter the VMFPSU command to create the output file *appid* **PSUPLAN**.

vmfpsu esa *compname*

2 Review the beginning of the *appid* **PSUPLAN** file for the sections that show the VMFPSUnnnnI messages. These messages will indicate the number of new PTFs on the RSU, the number of PTFs currently applied that will need to be reapplied, the number of PTFs to be excluded, and the number of local modifications affected by service on the RSU. This information will also be displayed on your console.

Decide which receive method is best for you. The Preferred Method: Receive the Preapplied, Prebuilt Service on page 2-11 or the Alternate Method: Receiving PTFs Only from the RSU Tape on page 2-11.

An example 2VMVMA40 PSUPLAN file for CMS could appear as:

```
*********************************************************************
****            PPFNAME: ESA         COMPNAME: CMS
*********************************************************************
****    PRODID: 2VMVMA40%CMS         Service Level: nnn-9901
*********************************************************************
****            Date: mm/dd/yy       Time: hh:mm:ss
*********************************************************************
VMFPSU1071I There are 10 PTFs on the Recommended
            Service Upgrade for PRODID 2VMVMA40%CMS
            that are not currently applied.
VMFPSU1072I There are 2 PTFs currently applied to
            PRODID 2VMVMA40%CMS that need to be
            reapplied.
VMFPSU1076I There are 0 PTFs to be excluded from the
            Recommended Service Upgrade.
VMFPSU1073I There are 2 parts with local modifications
            that need to be reprocessed.
VMFPSU1078I Select data file DMS$PSU$ was created
            or updated to force the rebuild of the local
            modifications.
*********************************************************************
****  PTFS TO BE APPLIED FOR PRODID 2VMVMA40%CMS
*********************************************************************
    PTF.APAR         PTF.APAR         PTF.APAR         PTF.APAR
UM11111.PM11111  UM12345.PM12345  UM22222.PM22222  UM33333.PM33333
UM44444.PM44444  UM55555.PM55555  UM66666.PM66666  UM77777.PM77777
UM88888.PM88888  UM99991.PM99991
*********************************************************************
****  PTFS TO BE REAPPLIED TO PRODID 2VMVMA40%CMS
*********************************************************************
   PTF.APAR          PTF.APAR         PTF.APAR         PTF.APAR
UM88889.PM88889  UM99997.PM99997
*********************************************************************
****  PTFS EXCLUDED FOR PRODID 2VMVMA40%CMS
*********************************************************************
NONE
*********************************************************************
****  LOCALMODS TO REPROCESS: 2VMVMA40 VVTLCL ON DISK 3C4(E)
*********************************************************************
PART - ABCDEF TXT
  PTF - UM12345.PM12345
  MOD - LCL0001.UPDT0001
*********************************************************************
****  LOCALMODS TO REPROCESS: 2VMVMA40 VVTLTST ON DISK 3C4(E)
*********************************************************************
PART - IJKLM EXC
  PTF - UM11111.PM11111  UM22222.PM22222
  MOD - LCL0002
```

Figure 2-3. Example 2VMVMA40 PSUPLAN

In this example, there are zero PTFs to be excluded (message VMFPSU1076I). Therefore, you can still choose between both methods. Message VMFPSU1071I indicates there are 10 PTFs to apply. Message VMFPSU1072I indicates there are 2 PTFs currently applied that will need to be reapplied. Because there are no PTFs to

exclude and there are fewer PTFs to reapply than apply, the Preferred Method is the best choice.

For more information about *appid* **PSUPLAN** files, see the *VM/ESA: VMSES/E Introduction and Reference*.

Receive a Component from the RSU

Note

Receive the component from the RSU by choosing substep 1, the Preferred Method, **or** substep 2, the Alternate Method. Then continue to substep 3 on page 2-12.

1 Preferred Method: Receive the Preapplied, Prebuilt Service

Refresh the selected component's service disks by loading new service from the RSU. The option OVERRIDE NO suppresses the prompt to create an override file.

If tape:

vmfins install ppf esa *compname* **(nomemo nolink override no**

If envelope:

vmfins install ppf esa *compname* **(nomemo nolink override no env rptf***num*

Use env rptf*num* if the RSU was received as an envelope file.

Continue on to substep 3.

2 Alternate Method: Receiving PTFs Only from the RSU Tape

Generate the ESAPTFS PPF file for the component to insure you have the most current level of PPF file.

vmfppf esaptfs *compname*

If you have made your own overrides, be sure to incorporate those in the ESAPTFS PPF file. For information on how to do this, see the *VM/ESA: VMSES/E Introduction and Reference*.

Receive **only** PTFs, for this component, from the RSU and not the tape files containing the preapplied and prebuilt objects. The option OVERRIDE NO suppresses the prompt to create an override file.

If tape:

vmfins install ppf esaptfs *compname* **(nomemo nolink override no**

If envelope:

vmfins install ppf esaptfs *compname* **(nomemo nolink override no env rptf***num*

Use ENV rptf*num* if the RSU was received as an envelope file.

3 Check the message log ($VMFINS $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview install

Process Additional Service

Note

Apply the component that was received from the RSU.

For preapplied, prebuilt service any reach-ahead service that may be on the system will be re-applied or there will be an indication that there are no reach-ahead PTFs to be applied. For PTFs only, the service that was received from the RSU and all PTFs in the apply list that are not currently applied will be applied.

1 If you chose the Alternate Method and you have PTFs that you want to exclude, add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

2 Apply additional service:

vmfapply ppf esa *compname* **(setup**

Messages VMFAPP2122E and VMFAPP2109R will be displayed only if you have reach-ahead service that needs to be reapplied.

```
VMFAPP2122E The set of PTFs in the apply status table (appid SRVAPPS ) on the
            nnn (fm-int) disk is not a subset of the PTFs in the highest level apply
            status table on the mmm (fm-alt) disk. This is an inconsistent state
VMFAPP2109R VMFAPPLY will automatically correct the problem identified by
            message 2122E by including the missing PTFs in the current apply
            list. Enter (1) to continue; (0) to quit.
1
```

If you receive these messages, enter **1** for VMFAPPLY to reapply the reach-ahead service.

3 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview apply

4 If necessary, rework ***local service***.

The output from the VMFPSU command, in the file *appid* **PSUPLAN**, can be used to indicate what local service is affected by the RSU. If a PTF is applied and it contains service to a part for which you have a local modification, you will need to rework the local service.

Figure 2-4 shows a section of a possible **PSUPLAN** file for CMS. Look for the heading LOCALMODS TO REPROCESS. Each heading will contain one or more local modifications that might have to be reworked.

```
*****************************************************************
****  LOCALMODS TO REPROCESS: 2VMVMA40 VVTLCL ON DISK 3C4(E)
*****************************************************************
PART - ABCDEFGH TXT
  PTF - UM12345.PM12345
  MOD - LCL0001.UPDT0001
PART - LMNOP EXC
@ PTF - UM22222.PM22222
  MOD - LCL0003
```

Figure 2-4. Section example of a PSUPLAN

Also in the figure, if PTF UM12345 is applied (because of APAR PM12345) then you would need to rework the local modification for LCL0001 for part ABCDEFGH TXT.

Another possible situation is that the local modification will be listed, but the PTF was marked for exclusion. This situation would be shown with an @ preceding the PTF line in the LOCALMODS TO REPROCESS section. Therefore, the local modification does not have to be reworked.

To rework local service follow the instructions in Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Follow the steps starting at the beginning of the chapter, but return here when you reach “Step 4h. Rebuild Remaining Objects” on page 7-9. The rebuild will be done in the next step.

5 If message VMFPSU1078I appears in the PSUPLAN file, a select data file has been updated to force the rebuild of all local service. If you are using the Preferred Method the necessary rebuilds will occur automatically during Build processing. If you are using the Alternate Method and you do not want these builds to occur, you must erase the $SELECT file that is identified in message VMFPSU1078I.

Build the New Service Level

Perform the build process for each component that has been serviced.

The build procedure for “Preapplied, Prebuilt Service” may require that you rebuild a segment or the CMS, CP, or GCS nuclei because the nuclei and segments are not shipped on the RSU. If there is no reach-ahead service, then all other objects or parts have been built and received to the component's test build disk during the receive of the RSU. If there is reach-ahead service, (for example, corrective service or local service that you had already applied to your system but is not on the RSU) the affected objects will be rebuilt.

The build procedure for “PTFs Only” will identify and build the objects for the PTFs that were received and applied from the RSU.

In both cases, objects affected by local service will be rebuilt.

Follow the steps defined for each component. Use the following table to locate the build procedure for each component. After the build task has been completed for the component continue with the next section “Service Another Component?.”

Table 2-2. Build Procedures

Component	**page**
VMSES/E	“Prepare to Build” on page 3-10
REXX/VM	“Prepare to Build” on page 3-17
CMS	“Prepare to Build” on page 3-24
CP	“Prepare to Build” on page 3-34
GCS	“Prepare to Build” on page 3-42
Dump Viewing Facility	“Prepare to Build” on page 3-51
TSAF	“Prepare to Build” on page 3-58
AVS	“Prepare to Build” on page 3-65

Note: If you have completed building the VMSES/E component then you should ACCESS the 5E6 disk as B to use the new service level when servicing other components. (You only need to do this now because you have not yet placed the service into production.)

Service Another Component?

You have now completed processing a component. If you have another component to service, go to “Prepare to Receive a Component from the RSU” on page 2-9.

Process the PSP Bucket COR Tape

If you received a Preventive Service Planning (PSP) bucket COR tape, go to Chapter 3, “Using the COR Service Procedure” on page 3-1 and follow the instructions for processing COR service. These instructions will eventually take you to “Step 11. Place Your Serviced Components into Production” on page 3-68. You will **not** return to this chapter (Chapter 2, “Using the Product Service Upgrade (PSU) Procedure” on page 2-1).

If you have not received a Preventive Service Planning (PSP) bucket COR tape, continue with the next section (“Place Product Service into Production” on page 2-15).

Place Product Service into Production

After all components have been serviced and tested, follow the instructions in the section “Step 11. Place Your Serviced Components into Production” on page 3-68 to rebuild saved segments and copy test build disks to production build disks.

If you have not already done so, see Appendix D, “Test the New Level of CMS” on page D-1 for a procedure to test CMS.

Note

You have now reached the end of the Product Service Upgrade procedure.

Chapter 3. Using the COR Service Procedure

This chapter contains the procedure to install corrective service to all components of VM/ESA. Corrective service is shipped on a corrective service (COR) tape.

Determine RSU Service Level Before Ordering COR Service

Use the VMFSIM command to determine the current RSU service level. In the example, 9901 is the RSU service level which would be used when ordering corrective service. The VM SYSRECS table contains a list of all products or components that are received on the system.

To query the status of all products or components installed using the ESA PPF file enter the following VMFSIM command (substitute ESAPTFS for ESA if that is the name of your PPF file):

```
| vmfsim query vm sysrecs tdata :ppf esa :stat
| VMFSIP2480I RESULTS FOR
|             TDATA :PPF ESA :STAT
| :PPF ESA VMSES
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9903
| :PPF ESA REXX
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9902
| :PPF ESA CMS
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9904
| :PPF ESA CP
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9904
| :PPF ESA GCS
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9902
| :PPF ESA DV
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9901
| :PPF ESA TSAF
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9901
| :PPF ESA AVS
|    :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.240-9901
| READY; T=s.ss/s.ss hh:mm:ss
```

The query returns the following data:

Field	Data Definition
:PPF	This field contains the file name of the PPF that was used to install the product or component, and the component name in the PPF that was used.
:STAT	This field contains the status of the product or component, followed by a number of qualifiers that identify the date and time that it was received, the user ID used to receive it, and the service level when it was installed. In this example, the 9901 is the RSU service level.

How to Install COR Service

You install COR service by component. There is a step in the service procedure for each component and each contains the specific commands and operands for the component.

To install COR service to VM/ESA, you must:

1. Prepare the system for service.
2. Receive the service documentation.
3. Service all needed VM/ESA components.
4. Place the newly-serviced components into production.

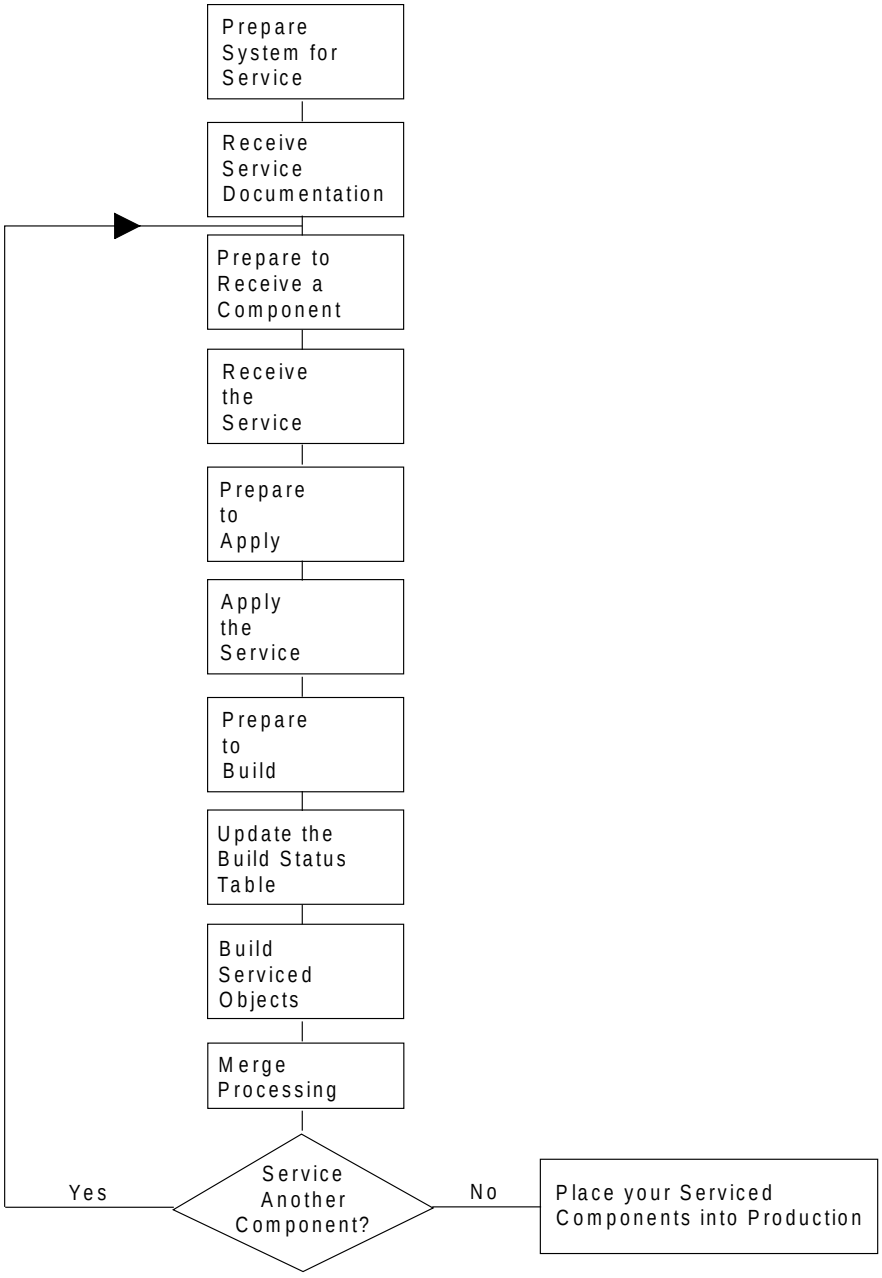

Figure 3-1. Service Application Flowchart using COR

To service each component, you must:

1. Receive the service for the component.
   a. Read the Memo-to-Users for the component.
   b. Use the VMFREC command to receive the service.
   c. Review the receive message log and correct any errors.
2. Apply the service to the component parts.
   a. Verify that the component's alternate apply disk is ready to apply service for the component and determine whether merge processing is required.

b. Use the VMFAPPLY command to apply the service.

c. Review the apply message log and correct any errors.

3. Rebuild the component.

   a. Perform any tasks that are required to rebuild the component.

   b. Review the build message log and correct any messages.

   c. Test the new level of the component and correct any errors.

To put serviced components into production:

1. Rebuild various segments and the CMS saved system.
2. Place the CP nucleus, CMS, TSAF, AVS and GCS into production.
3. Merge the intermediate apply disk, with the tested service, to the production service disk.

The receive and apply steps are similar for each component. The build step, however, varies from component to component. Building a component may involve rebuilding a nucleus, load libraries, or executable modules. Or it may only require replacing updated objects (such as EXECs and HELP files) onto a test build disk. For each component, you must decide whether to build each serviced object individually, or to build all objects at once. After all of the components are built, you must test the components and place them into production.

The order in which you service the VM/ESA components is important, because some components have dependencies on others and all of the procedures require the VMSES/E component to be at the latest level. For example, REXX/VM is serviced before CMS and GCS because it is needed to build the CMS, GCS or both nuclei correctly.

Complete the installation of service for each component before you proceed to the next. The VM/ESA components should be serviced in the following order:

1. VMSES/E
2. REXX/VM
3. CMS
4. CP
5. GCS
6. Dump Viewing Facility
7. TSAF
8. AVS.

Using Substitute Disks for Default Disks

Many of the commands used in the service procedure steps require you to enter the file mode of a disk. If you have changed any of the default disk addresses, you must substitute the file mode of your disk address for the file mode of the default disk address described in that step.

Spooling your Console for non-VMSES messages

All significant VMSES/E processing and messages will be logged in the appropriate message log. VMSES/E, however, is unable to capture and log other messages that might be issued, for example CMS and CP messages. For this reason spool your console to save the message output.

```
spool console start to *
```

Enter this command to begin spooling your console.

When you are ready to discontinue console spooling and save the console data as a reader spool file issue the command:

spool console stop close

Or, to discontinue console spooling and discard the console data issue this command:

spool console stop purge

The COR Service Procedure

The COR service procedure follows. **Follow the steps in order** starting with "Step 1. Prepare the System for Service."

Step 1. Prepare the System for Service

You need to prepare the system for service each time you apply new service to your system. However, if you install more than one COR tape at a time you need to prepare the system for service only once.

1 Log on MAINT.

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

2 As a precaution, do a complete backup of your system. See the *VM/ESA: Installation Guide* for instructions on storing a backup copy of the VM/ESA system to tape.

3 Set up the correct virtual machine environment and IPL.

```
set machine xa
System reset.
System = XA
```

If your machine environment is already in the correct mode, you will not see all of these displayed messages.

define storage *nn***M**
STORAGE CLEARED - SYSTEM RESET
STORAGE = nnM

Define the storage to at least 35MB. If you will be servicing GCS, make sure to define your storage to be at least as large as the value you defined for HIGH COMMON END storage (default 19MB) during the installation procedures. For more information, see *VM/ESA: Installation Guide*.

Defining storage causes a system reset.

ipl 190 clear

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

ENTER

Press ENTER to complete the IPL.

4 Access the required disks for the service procedure, if necessary.

access *vdev* **a**

You must have a R/W A-disk accessed.

access 5e5 b

Access the VMSES/E installation and service tools disk.

access 51d d

Access the software inventory disk.

release c

If you do not want to load the service documents to the C-disk, release the C-disk.

5 If you want to change the IBM service defaults, change the PPFs using $PPF overrides on the VMSES/E system inventory disk (51D) to reflect your system configuration. For information about tailoring the PPF, see *VM/ESA: VMSES/E Introduction and Reference*.

Note

If you have updated the 190 or 193 disks you should copy any new and changed files to the 490 and 493 disks. This will avoid accidentally back-leveling modified files.

Step 2. Receive the Service Documentation

Note

This step describes how to receive service documentation from a service tape. If you have received service from ServiceLink that now resides on disk, use “Receive Service Documentation” on page A-1 to receive service documentation, instead of this step. Then return to “Step 3. Service VMSES/E” on page 3-7.

1 Attach a tape drive to your virtual machine (MAINT).

attach *rdev* *** 181**

rdev is the real device number of the tape drive you want to attach.

2 Mount and ready the COR tape.

3 Receive the documentation. VMFREC with the INFO option loads the documentation and displays a list of all products on the tape.

```
vmfrec info
```

4 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use on-line HELP.

vmfview receive

Also, make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the components on the tape.

5 Read the product memo(s) (*prodid* MEMO) before going on. These memos are loaded to the C-disk if it is accessed R/W; otherwise, the memos are loaded to the A-disk which is accessed R/W in “Step 1. Prepare the System for Service” on page 3-5.

Step 3. Service VMSES/E

Note

If you do not have VMSES/E service on the tape, go to “Step 4. Service REXX/VM” on page 3-14.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMK40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **vmses**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **vmses**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your own PPF override file.

```
vmfppf ppfname compname
```

5 Set up your access order.

```
vmfsetup esa vmses
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use “Receive the Service” on page A-2 to receive service. Then return to the next section “Prepare to Apply Service” in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in “Step 2. Receive the Service Documentation” on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa vmses
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa vmses
```

2 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa vmses
```

3 Enter the VMFMRDSK command to clear the VMSES/E alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level.

```
vmfmrdsk esa vmses apply
```

4 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa vmses
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview apply

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
********************************************************************************
****    PPFNAME: ESA        COMPNAME: VMSES         APPID: 2VMVMK40 ****
********************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
********************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for VMSES/E, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If your PPF file has changed since you performed the last task for this component step, enter the following command:

vmfppf esa vmses

2 If your access order has changed since you performed the last task in this component step, enter the following command:

vmfsetup esa vmses

Update the Build Status Table

1 Update the Build Status Table with serviced parts.

vmfbld ppf esa vmses (status

This command updates the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMK40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMK40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa vmses
copy 2vmvmk40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmk40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa vmses

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa vmses (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and review what objects need to be built:

vmfview build

Build Serviced Objects

Note

Use the remainder of this section to build all VMSES/E objects that were serviced. If you prefer to build VMSES/E objects individually, then use “Detailed Build Steps: VMSES/E” on page 4-1. Then return to “Rebuild the HELP Files in Uppercase (Optional).”

1 To rebuild all serviced parts, enter the following command:

```
vmfbld ppf esa vmses (serviced
```

Note: If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn ft have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

3 Use the new serviced level of VMSES/E throughout the rest of this procedure by entering the following command:

```
access 5e6 b
```

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

```
vmfppf uceng vmses
```

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

```
vmfbld ppf uceng vmses vmfblhlp (serviced setup
```

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate apply disk, and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

You can ignore the VMFSET2206W warning message that may be received when you enter the command. The message is displayed because the 5E6 disk is already accessed.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level.

```
vmfmrdsk esa vmses apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, "Using the COR Service Procedure" on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• "Step 11. Place Your Serviced Components into Production" on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	"Service Another Component?" on page 2-14 to receive service for the next component.

Step 4. Service REXX/VM

Note

If you do not have REXX/VM service on the tape, go to "Step 5. Service CMS" on page 3-20.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMF40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **rexx**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **rexx**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in "Step 1. Prepare the System for Service" on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see "Step 2. Receive the Service Documentation" on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa rexx
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use "Receive the Service" on page A-2 to receive service. Then return to the next section "Prepare to Apply Service" in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in "Step 2. Receive the Service Documentation" on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa rexx
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa rexx
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa rexx
```

4 Enter the VMFMRDSK command to clear the REXX/VM alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa rexx apply
```

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa rexx
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview apply

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
*************************************************************************
****    PPFNAME: ESA        COMPNAME: REXX        APPID: 2VMVMF40      ****
*************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
*************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLIXX which is a
CK:            higher level than VVTIXX.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLIXX = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for REXX/VM, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

access 5e6 b

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

vmfppf esa rexx

3 If your access order has changed since you performed the last task in this component step, enter the following command:

vmfsetup esa rexx

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

vmfbld ppf esa rexx (status

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMF40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMF40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa rexx
copy 2vmvmf40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmf40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa rexx

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa rexx (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be built. If a MACLIB needs to be re-built, ensure that your A-disk has enough free space to hold two copies of the MACLIB.

```
vmfview build
```

Build Serviced Objects

Note

Use the remainder of this section to build all REXX/VM objects that were serviced. If you prefer to build REXX/VM objects individually, then use “Detailed Build Steps: REXX/VM” on page 4-3. Then return to “Merge Processing.”

1 To rebuild all serviced parts, enter the following command:

```
vmfbld ppf esa rexx (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Nuclei

Note

Parts of the REXX/VM component are contained in the CMS nucleus and the GCS nucleus. You must receive, apply, and build any REXX/VM service before you build either nucleus. If you have applied REXX/VM service and received message VMFBDU2180I to rebuild CMS and GCS nuclei, then you must rebuild and IPL the CMS nucleus and, if you have installed GCS, the GCS nucleus. If you are going to service CMS or GCS, then do not rebuild that nucleus now because it will be rebuilt when that component is serviced. If there is no service to CMS or GCS, then rebuild and IPL its nucleus now. For information on rebuilding and IPLing the CMS nucleus, see “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15. For information on rebuilding and IPLing the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27. Then return to “Merge Processing.”

Merge Processing

1 Clear the alternate apply disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are

ready to commit it to this service level. For components that share disks, like REXX/VM and CMS, doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa rexx apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, "Using the COR Service Procedure" on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• "Step 11. Place Your Serviced Components into Production" on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	"Service Another Component?" on page 2-14 to receive service for the next component.

Step 5. Service CMS

Note

If you do not have CMS service on the tape, but you had service to the IXXBLNUC build list when servicing REXX/VM, and did not already rebuild the CMS nucleus, go to "Rebuild the CMS Nucleus" on page 4-13. Complete the rebuild and the IPL of the CMS nucleus. If you do not have CMS and REXX/VM service, skip this step, and go to "Step 6. Service CP" on page 3-31.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMA40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **cms**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **cms**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa cms
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use “Receive the Service” on page A-2 to receive service. Then return to the next section “Prepare to Apply Service” in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in “Step 2. Receive the Service Documentation” on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa cms
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa cms
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa cms
```

4 Enter the VMFMRDSK command to clear the CMS alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

vmfmrdsk esa cms apply

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

vmfapply ppf esa cms

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview apply

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
*************************************************************************
****    PPFNAME: ESA        COMPNAME: CMS          APPID: 2VMVMA40  ****
*************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
*************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for CMS, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

```
vmfppf esa cms
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa cms
```

Note

Parts of the REXX/VM component are contained in the CMS nucleus. You must receive, apply and build REXX/VM service (if any) before you build the CMS nucleus.

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

```
vmfbld ppf esa cms (status
```

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 on page 3-25 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMA40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMA40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa cms
copy 2vmvma40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvma40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa cms

Enter VMFSETUP to ensure correct access order.

vmfbld ppf esa cms (status
1

Enter the VMFBLD command to complete updating the build status table When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be rebuilt. If a MACLIB needs to be re-built, ensure that your A-disk has enough free space to hold two copies of the MACLIB.

vmfview build

Note: If you have VSE/VSAM installed, determine if the CMSVSAM DOSLIB and the CMSVSAM segment need to be rebuilt.

Use VMFVIEW to view the build messages. If DMSVIP.TEXT in build list DMSBL493 has a status of SERVICED, then the CMSVSAM DOSLIB and segment will have to be rebuilt. See the Program Directory for VSE/VSAM for instructions on how to service VSE/VSAM. Note you should finish servicing CMS and put the service into production before you service VSE/VSAM.

Build Serviced Objects

Note

Use the remainder of this section to build all CMS objects that were serviced. If you prefer to build CMS objects individually, then use “Detailed Build Steps: CMS” on page 4-5. Then return to “Test the New Level of CMS” on page 3-30.

1 If you make any changes to the DMSNGP ASSEMBLE file, you must reassemble the file. First check the status of the DEFNUC macro. If it has

been serviced, then the DMSOM MACLIB should be rebuilt before reassembling DMSNGP. To check the status enter the following command:

```
vmfsim query 2vmvma40 srvblds * tdata :object defnuc :stat
VMFSIP2480I RESULTS FOR
            TDATA :OBJECT DEFNUC
:BLDLIST DMSOM
   :OBJECT DEFNUC
   :STAT SERVICED.06/28/93.13:09:59.MAINT
```

If the STAT field indicates that DEFNUC has been serviced see "Update the Macro Libraries (MACLIBs)" on page 4-8 for instructions to update the DMSOM MACLIB.

vmf-assemble **dmsngp esa cms ($select outmode localmod logmod**

vmf-assemble can be the VMFASM, VMFHASM, or VMFHLASM command depending on which assembler you have on your system.

Using these options, DMSNGP TXT will be placed into your $SELECT file. The DMSNGP TXT*modid* file will also be placed on your LOCALMOD disk and the local VVT will be updated if necessary.

For more information about the DMSNGP ASSEMBLE file, see the *VM/ESA: Installation Guide* and *VM/ESA: Planning and Administration* book.

2 Spool the printer and punch:

```
spool prt *
spool pun *
```

Spool the printer and punch output to your virtual reader. This is required, to have the serviced nucleus transferred to your reader.

Note for CMS Pipelines Users

If you have DB2 Server for VM* or ISPF or both, and are using the CMS Pipelines interfaces to these applications you must enter the GLOBAL command to identify the text libraries containing the interface files before running VMFBLD.

`global txtlib pipsql`

pipsql is the file name of the TXTLIB containing the DB2 Server for VM interface files, ISPF interface files, or both.

For more information on how the TXTLIB is created and where it is located, see the *VM/ESA: Planning and Administration*.

3 To rebuild all serviced parts, enter the following command:

```
vmfbld ppf esa cms (serviced
```

Notes:

a. If you receive the following message:

```
DMSLIO201W The following names are undefined: ISPLINK ARIPRDI
```

disregard it, because they are interface files for ISPF and SQL (if you are not using these as interfaces into DMSPIPE).

b. If, while building the DMSPIPE MODULE, you get a return code of 4 with this message:

```
DMSLIO201W The following name is undefined: DMSDSCSC
```

disregard it, because the undefined name is a user exit.

c. If you receive the following message while building the CSL libraries:

```
DMSSOP2534I No MACLIB libraries have been globaled.
```

you can disregard it

d. If you receive the following message:

```
VMFSBR2000I  Objects in segment build list fn ft
             have been built or deleted.  Any segments
             using this build list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

4 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use on-line HELP.

```
vmfview build
```

Note for NLS

If you would like to change the default system national language for CMS you must rebuild the CMS nucleus to contain the language. To do this, follow the steps in Appendix C, “Building the NLS Nuclei, Including UCENG” on page C-1. This appendix will send you to “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15. Then return here to “Test the New Level of CMS” on page 3-30.

IPL the CMS Nucleus

1 It is possible the nucleus was rebuilt by the VMFBLD command due to CMS or REXX/VM service.

Query your reader to determine if you have a new nucleus. If you do, continue; if you do not then skip this section and continue with “Test the New Level of CMS” on page 3-30.

```
query rdr * all ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME     ...
MAINT    0025 A PUN 00043417 001 NONE 05/30 14:39:14 $$$TLL$$ ...
```

2 Make the file that contains the new nucleus the first file in your reader. If you have two copies order the reader so that the most recent one is first.

order rdr *spoolid*

spoolid is the spool ID of the nucleus file that you recorded earlier. In the previous example, the spool ID was 0025.

3 IPL the new nucleus.

define 190 590

Ensure that the nucleus gets loaded to the CMS test system disk, and does not overlay your production nucleus.

define 490 190

Note

The 490 disk is defined as 190 because the default address in DMSNGP for the nucleus is 190. If you have changed DMSNGP, substitute your address for the 190 in the two commands.

ipl 00c clear

Note: At this point, you might receive one or more nucleus generation prompts. A prompt appears if a **?** appears as an entry in, or if information is missing from the DMSNGP ASSEMBLE file. For more information about how to respond to these prompts, see the discussion of the DMSNGP profile in the chapter “Tailoring CMS with the DEFNUC macro” in the *VM/ESA: Planning and Administration* manual.

```
HCPNSS446E The Named Saved System (NSS) CMS was not
           previously defined and cannot be saved.
```

If you changed the SAVESYS operand in DMSNGP ASSEMBLE to SAVESYS=YES you will receive message HCPNSS446E. You can ignore this message. The CMS named saved system will be saved in “Step 11. Place Your Serviced Components into Production” on page 3-68.

```
VM/ESA V2.4.0 mm/dd/yy hh:mm
```

If you defined your own version ID, it appears here and each time you IPL the CMS test system disk (MAINT 490).

ENTER

Press ENTER to complete the IPL.

4 Save the CMS load map.

access 493 *fm-493*

fm-493 is any free file mode.

spool prt nohold
close prt

```
RDR FILE 0026 SENT FROM MAINT PRT
    WAS 0026 RECS 8270 CPY  001 A NOHOLD NOKEEP
```

The CMS load map has been spooled to your virtual reader. Note the spool ID for use in the next command.

receive *spoolid* **cmsnuc map** *fm-493* **(replace**

Receive the file onto the system tools disk, using the spool ID from the previous message. In the previous example, the spool ID was 0026.

vmfcopy cmsnuc map *fm-493* **= = = (prodid 2vmvma40%cms replace pack**

5 XEDIT the CMSNUC MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild and IPL the nucleus, see “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15. Then return to “Test the New Level of CMS” on page 3-30.

xedit cmsnuc map
====> set case mixed ignore
====> all/unresol/

```
DMSXDC546E Target not found
```

This response means that no unresolved symbols were found.

====> all/undefin/

```
DMSXDC546E Target not found
```

This response means that no undefined symbols were found.

====> qquit

6 Redefine the CMS disks back to their original values:

```
define 190 490
define 590 190
ipl 190 clear
VM/ESA V2.4.0 mm/dd/yy hh:mm
```

7 Access the necessary VMSES/E disks:

```
access address b
access 51d d
```

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

Test the New Level of CMS

The VM Callable Services Library (VMLIB), Shared File System (SFS) modules, REXX/VM component, and CMS nucleus are all functionally dependent on each other. For a minimal test of CMS to ensure that these functional dependencies are maintained after installing new service see Appendix D, “Test the New Level of CMS” on page D-1. Then continue with “Rebuild the HELP Files in Uppercase (Optional).”

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

```
vmfppf uceng cms
```

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

```
vmfbld ppf uceng cms dmsblhlp (serviced setup
```

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate apply disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level.

For components that share disks, like CMS and REXX/VM, doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa cms apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use on-line HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, "Using the COR Service Procedure" on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• "Step 11. Place Your Serviced Components into Production" on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	"Service Another Component?" on page 2-14 to receive service for the next component.

Step 6. Service CP

Note

If you do not have CP service on the tape, go to "Step 7. Service GCS" on page 3-39.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMB40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **cp**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **cp**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa cp
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use “Receive the Service” on page A-2 to receive service. Then return to the next section “Prepare to Apply Service” in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in “Step 2. Receive the Service Documentation” on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa cp
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa cp
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa cp
```

4 Enter the VMFMRDSK command to clear the CP alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa cp apply
```

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa cp
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview apply
```

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
********************************************************************************
****    PPFNAME: ESA        COMPNAME: CP          APPID: 2VMVMB40   ****
********************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
********************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for CP, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here. If the HCPMDLAT MACRO was affected, the instructions for updating the CP Load List are in Appendix F, “Updating the CP Load List” on page F-1.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

```
vmfppf esa cp
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa cp
```

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

```
vmfbld ppf esa cp (status
```

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 on page 3-36 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMB40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

```
0
```

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMB40 $PPF on 191 (A) from level $PFnnnnn
```

```
vmfppf esa cp
```

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

copy 2vmvmb40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmb40 $ppf a

fm-51d is the file mode of the software inventory disk.

For the CP component, there might be a list of other $PPF files that will also need to be copied. (The names will be displayed in the message.) If this happens, repeat the COPY and ERASE command for each $PPF file.

vmfsetup esa cp

Enter VMFSETUP to ensure correct access order.

vmfbld ppf esa cp (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be rebuilt: If a MACLIB needs to be re-built, ensure that your A-disk has enough free space to hold two copies of the MACLIB.

vmfview build

Build Serviced Objects

Note

Use the remainder of this section to build all CP objects that were serviced. If you prefer to build CP objects individually, then use “Detailed Build Steps: CP” on page 4-17. Then return to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-38.

1 To rebuild all serviced parts, enter the following command:

vmfbld ppf esa cp (serviced

Note:

If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn fm have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Stand-Alone Dump

1 Use VMFVIEW to see if the object SA.DUMP was serviced and requires additional build steps.

vmfview build
BD:VMFBDU2180I HCPBLUTL SA.DUMP RUN HCPSADMP EXEC

If this message is shown then you need to rebuild the stand-alone dump.

2 If necessary, use the HCPSADMP EXEC to rebuild the Stand-Alone Dump and then re-IPL CMS.

hcpsadmp
ipl 190 clear
VM/ESA V2.4.0 *mm/dd/yy hh:mm*

HCPSADMP loads a Disabled Wait State PSW when it completes. For more information about this exec, see *VM/ESA: Planning and Administration*.

ENTER

Press ENTER to complete the IPL.

3 Access the necessary disks:

access *address* **b**
access 493 *fm-493*
access 51d d

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

Note for NLS

If you would like to change the default system national language for CP you must rebuild the CP nucleus to contain the language. To do this, follow the steps in Appendix C, “Building the NLS Nuclei, Including UCENG” on page C-1. This appendix will send you to “Rebuild the CP Nucleus” on page 4-21 and “Verify the CP Load Map” on page 4-22. Then return here to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-38.

Verify the CP Load Map

1 XEDIT the CPLOAD MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild the CP nucleus and save the map, see “Rebuild the CP Nucleus” on page 4-21 and Verify the CP Load Map. Then continue with “Rebuild the HELP Files in Uppercase (Optional)” on page 3-38.

xedit cpload map *fm-493*
====> set case mixed ignore

fm-493 is the file mode of the target disk which is where the CPLOAD MAP was placed when it was created.

====> all/unresol/
DMSXDC546E Target not found

This response means that no unresolved symbols were found.

====> all/undefin/
DMSXDC546E Target not found

This response means that no undefined symbols were found.

====> qquit

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

vmfppf uceng cp

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

vmfbld ppf uceng cp hcpblhlp (serviced setup

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate apply disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service to the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks, like CP and Dump Viewing Facility, doing a merge on one component will merge all the files from the other component.

vmfmrdsk esa cp apply (setup

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

vmfview merge

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, "Using the COR Service Procedure" on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• "Step 11. Place Your Serviced Components into Production" on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	"Service Another Component?" on page 2-14 to receive service for the next component.

Step 7. Service GCS

Note

1. If you do not have GCS service on the tape, but you had service to the IXXBLNUC build list when you serviced REXX/VM, go to "Rebuild the GCS Nucleus" on page 4-26 and then "IPL the GCS Nucleus" on page 4-27. If you do not have GCS and REXX/VM service, skip this step and go to "Step 8. Service the Dump Viewing Facility" on page 3-48.
2. When entering the component name, or *compname* in this procedure, if you installed GCS in the Shared File System, use **gcssfs** as the *compname*, instead of **gcs**. Moving GCS from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.
3. The notation for the component name is **gcs|gcssfs**. Enter **gcs** or **gcssfs**. Do not enter the **|**.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVML40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **gcs|gcssfs**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **gcs|gcssfs**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa gcs|gcssfs
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use “Receive the Service” on page A-2 to receive service. Then return to the next section “Prepare to Apply Service” in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in “Step 2. Receive the Service Documentation” on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa gcs|gcssfs
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa gcs|gcssfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa gcs|gcssfs
```

4 Enter the VMFMRDSK command to clear the GCS alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa gcs|gcssfs apply
```

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa gcs|gcssfs
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview apply
```

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
********************************************************************************
****    PPFNAME: ESA        COMPNAME: GCS         APPID: 2VMVML40   ****
********************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
********************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for GCS, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa gcs|gcssfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa gcs|gcssfs
```

4 Make any other changes that you want to include in the nucleus, such as changes to the GCS configuration file using the GROUP EXEC. For more information, see *VM/ESA: Installation Guide*.

Note

Parts of the REXX/VM component are contained in the GCS nucleus. You must receive and apply REXX/VM service (if any) before you rebuild the GCS nucleus.

5 Spool the printer and punch for the nucleus build.

```
spool prt *
spool punch *
```

Direct the output of the virtual printer and punch to your virtual reader.

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

```
vmfbld ppf esa gcs|gcssfs (status
```

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 on page 3-44 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVML40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0 — Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVML40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa gcs|gcssfs
copy 2vmvml40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvml40 $ppf a — Enter the VMFPPF command to ensure you have the appropriate level of PPF. *fm-51d* is the file mode of the software inventory disk.

vmfsetup esa gcs|gcssfs — Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa gcs|gcssfs (status
1 — Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be rebuilt: If a MACLIB needs to be re-built, ensure that your A-disk has enough free space to hold two copies of the MACLIB.

vmfview build

Note

If the results of the VMFVIEW command show that the GCTLOAD build list was serviced, then the new GCS nucleus build list(s) added during Appendix J in the *VM/ESA: Installation Guide* will need to be built. After issuing the VMFBLD command during step 2 on page 3-45 in "Build Serviced Objects" issue the following VMFBLD command:

vmfbld ppf esa gcs|gcssfs *bldlist* (all where *bldlist* is the build list name.

Build Serviced Objects

Note

Use the remainder of this section to build all GCS objects that were serviced. If you prefer to build GCS objects individually, then use "Detailed Build Steps: GCS" on page 4-23. Then return to "Rebuild the HELP Files in Uppercase (Optional)" on page 3-47.

1 To rebuild all serviced parts, enter the following command:

vmfbld ppf esa gcs|gcssfs (serviced

Note:

If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn ft have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in "Step 11. Place Your Serviced Components into Production" on page 3-68.

2 Use VMFVIEW to review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use on-line HELP.

vmfview build

Note for NLS

If you would like to change the default system national language for GCS you must rebuild the GCS nucleus to contain the language. To do this, follow the steps in Appendix C, "Building the NLS Nuclei, Including UCENG" on page C-1. This appendix will send you to "Rebuild the GCS Nucleus" on page 4-26 and "IPL the GCS Nucleus" on page 4-27. Then return here to "Rebuild the HELP Files in Uppercase (Optional)" on page 3-47.

IPL the GCS Nucleus

1 It is possible the nucleus was rebuilt due to GCS or REXX/VM service.

Query your reader to determine if you have a new nucleus. If you do, continue; if you do not then skip this section and continue with "Rebuild the HELP Files in Uppercase (Optional)" on page 3-47.

```
query rdr * all
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME     ...
MAINT    0025 A PUN 00043417 001 NONE 05/30 14:39:14 $$$TLL$$ ...
```

2 Make the file that contains the new nucleus the first file in your reader and IPL the new nucleus.

Note: This will put GCS into production.

order rdr *spoolid*

spoolid is the spool ID of the nucleus file that you recorded earlier. In the previous example, the spool ID is 0025.

ipl 00c clear

Save the nucleus in your saved segment.

```
NAMED SAVED SYSTEM (NSS) GCS
WAS SUCCESSFULLY SAVED IN FILEID 0100
```

This message indicates that your nucleus was successfully saved.

```
RDR FILE spoolid SENT FROM MAINT
```

Record the spool file ID from this message. It identifies the file containing the nucleus load map that was spooled to your virtual reader.

ipl 190 clear
VM/ESA V2.4.0 *mm/dd/yy hh:mm*

Restore your CMS environment.

ENTER

Press ENTER to complete the IPL.

3 Access the necessary VMSES/E disks:

access *address* **b**
access 51d d

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

4 Save the load map on the test system tools disk (493).

access 493 *fm-493*

fm-493 is any file mode for the 493 disk.

receive *spoolid* **gcsnuc map** *fm-493* **(olddate replace**

spoolid is the spool file ID of the GCS load map that was sent to your reader when you built the GCS nucleus.

vmfcopy gcsnuc map *fm-493* **= = = (prodid 2vmvml40%gcs replace pack**

5 XEDIT the GCSNUC MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild and IPL the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27. Then return to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-47.

xedit gcsnuc map
====> set case mixed ignore
====> all/unresol/

```
DMSXDC546E Target not found
```

This response means that no unresolved symbols were found.

====> all/undefin/
DMSXDC546E Target not found

This response means that no undefined symbols were found.

====> qquit

release *fm-493*

fm-493 is the file mode of the 493 disk.

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

vmfppf uceng gcs|gcssfs

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

vmfbld ppf uceng gcs|gcssfs gctblhlp (serviced setup

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level.

vmfmrdsk esa gcs|gcssfs apply (setup

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use on-line HELP.

vmfview merge

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, “Using the COR Service Procedure” on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• “Step 11. Place Your Serviced Components into Production” on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	“Service Another Component?” on page 2-14 to receive service for the next component.

Step 8. Service the Dump Viewing Facility

Note

If you do not have Dump Viewing Facility service on the tape, go to “Step 9. Service TSAF” on page 3-54.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMI40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **dv**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **dv**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa dv
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use "Receive the Service" on page A-2 to receive service. Then return to the next section "Prepare to Apply Service" in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in "Step 2. Receive the Service Documentation" on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa dv
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa dv
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa dv
```

4 Enter the VMFMRDSK command to clear the Dump Viewing Facility alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa dv apply
```

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa dv
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview apply

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
*************************************************************************
****    PPFNAME: ESA        COMPNAME: DV          APPID: 2VMVMI40   ****
*************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
*************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for Dump Viewing Facility, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

access 5e6 b

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

vmfppf esa dv

3 If your access order has changed since you performed the last task in this component step, enter the following command:

vmfsetup esa dv

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

vmfbld ppf esa dv (status

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMI40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMI40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa dv
copy 2vmvmi40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmi40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa dv

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa dv (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be rebuilt:

```
vmfview build
```

Build Serviced Objects

Note

Use the remainder of this section to build all Dump Viewing Facility objects that were serviced. If you prefer to build Dump Viewing Facility objects individually, then use “Detailed Build Steps: Dump Viewing Facility” on page 4-28. Then return to “Rebuild the HELP Files in Uppercase (Optional).”

1 To rebuild all serviced parts, enter the following command:

```
vmfbld ppf esa dv (serviced
```

Note:

If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn ft have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

```
vmfppf uceng dv
```

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

```
vmfbld ppf uceng dv hcsblhlp (serviced setup
```

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks, like Dump Viewing Facility and CP, doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa dv apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, “Using the COR Service Procedure” on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• “Step 11. Place Your Serviced Components into Production” on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	“Service Another Component?” on page 2-14 to receive service for the next component.

Step 9. Service TSAF

Note

- If you do not have TSAF service on the tape, go to “Step 10. Service AVS” on page 3-61.
- When entering the component name, or *compname* in this procedure, if you installed TSAF in the Shared File System, use **tsafsfs** as the *compname*, instead of **tsaf**. Moving TSAF from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.
- The notation for the component name is **tsaf|tsafsfs**. Enter **tsaf** or **tsafsfs**. Do not enter the **|**.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMH40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **tsaf|tsafsfs**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **tsaf|tsafsfs**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

vmfppf *ppfname compname*

6 Set up your access order.

```
vmfsetup esa tsaf|tsafsfs
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use "Receive the Service" on page A-2 to receive service. Then return to the next section "Prepare to Apply Service" in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in "Step 2. Receive the Service Documentation" on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa tsaf|tsafsfs
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa tsaf|tsafsfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa tsaf|tsafsfs
```

4 Enter the VMFMRDSK command to clear the TSAF alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa tsaf|tsafsfs apply
```

now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa tsaf|tsafsfs
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview apply
```

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
********************************************************************************
****    PPFNAME: ESA        COMPNAME: TSAF        APPID: 2VMVMH40   ****
********************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
********************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for TSAF, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note: If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

```
vmfppf esa tsaf|tsafsfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa tsaf|tsafsfs
```

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

vmfbld ppf esa tsaf|tsafsfs (status

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMH40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMH40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa tsaf|tsafsfs
copy 2vmvmh40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmh40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa tsaf|tsafsfs

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa tsaf|tsafsfs (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be built:

vmfview build

Build Serviced Objects

Note

Use the remainder of this section to build all TSAF objects that were serviced. If you prefer to build TSAF objects individually, then use “Detailed Build Steps: TSAF” on page 4-31. Then return to “Rebuild the HELP Files in Uppercase (Optional).”

1 To rebuild all serviced parts, enter the following command:

vmfbld ppf esa tsaf|tsafsfs (serviced

Note:

If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn ft have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

vmfppf uceng tsaf|tsafsfs

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

vmfbld ppf uceng tsaf|tsafsfs atsblhlp (serviced setup

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate apply disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks, like TSAF and AVS, doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa tsaf|tsafsfs apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, “Using the COR Service Procedure” on page 3-1,	
• and this is not the last component you are servicing, then go to	• the next Step.
• and this is the **last** component that you are servicing, then go to	• “Step 11. Place Your Serviced Components into Production” on page 3-68 to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	“Service Another Component?” on page 2-14 to receive service for the next component.

Step 10. Service AVS

Note

- If you do not have AVS service on the tape, you have finished installing service for VM/ESA components.
- When entering the component name, or *compname* in this procedure, if you installed AVS in the Shared File System, use **avssfs** as the *compname*, instead of **avs**. Moving AVS from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.

  When issuing the VMFBLD command, use the *compname* **avs** or **avssfs** if you have a GCS/XA environment.
- The notation for the component name is **avs|avssfs**. Enter **avs** or **avssfs**. Do not enter the **|**.

Prepare to Receive Service

The *ppfname* is **esa**, if you are using the override PPF supplied by IBM for VM/ESA. It contains overrides to the base PPF file for this component, 2VMVMD40 $PPF. If you have your own PPF override file for this component, you should use your file's *ppfname*. The *ppfname* you use should be used **throughout** the rest of this step instead of **esa**.

The *compname* is **avs|avssfs**, if you are using the override component name within the VM/ESA PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The *compname* you use should be used **throughout** the rest of this step instead of **avs|avssfs**.

1 Make sure that a tape drive is attached to your virtual machine, and that the COR tape is mounted and ready, unless you are receiving service from disk.

2 Make sure that you have prepared the system for service.

If you have not already done the activities in “Step 1. Prepare the System for Service” on page 3-5, do so now before continuing.

3 Make sure that you have the service documentation you need to install service for this component.

If you have not already received the documentation for the service you are about to perform, do so now. For more information, see “Step 2. Receive the Service Documentation” on page 3-6.

4 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

5 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

```
vmfppf ppfname compname
```

6 Set up your access order.

```
vmfsetup esa avs|avssfs
```

Receive the Service

Note

1. If you are receiving service from disk (envelope file), you should use "Receive the Service" on page A-2 to receive service. Then return to the next section "Prepare to Apply Service" in this Step. Otherwise, use this procedure to receive service from tape.
2. Also, if you are installing multiple service tapes, you can receive all of the service for this component before applying and building it.

For **each** service tape you want to receive, do the following:

1 Mount and ready the service tape. Complete all steps as described in "Step 2. Receive the Service Documentation" on page 3-6.

2 Receive the service. VMFREC loads service onto the delta disk as defined in the PPF for this component.

```
vmfrec ppf esa avs|avssfs
```

3 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview receive
```

Prepare to Apply Service

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task in this component step, enter the following command:

```
vmfppf esa avs|avssfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa avs|avssfs
```

4 Enter the VMFMRDSK command to clear the AVS alternate apply disk:

The alternate apply disk will already be clear if merge processing was performed during the build step the last time service was applied. This step will merge the current service level on the alternate apply disk into the test service level on the intermediate apply disk.

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks (REXX/VM with CMS, CP with DV, and TSAF with AVS), doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa avs|avssfs apply
```

5 If you have PTFs that you want to exclude, you should add them now to your exclude list. For instructions about updating your exclude list, see *VM/ESA: VMSES/E Introduction and Reference*.

Apply the Service

1 Apply the service.

```
vmfapply ppf esa avs|avssfs
```

2 Check the apply message log ($VMFAPP $MSGLOG) for warning and error messages.

If necessary, correct any problems before going on. For information about handling specific apply messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview apply
```

Note that an RO message indicates an out-of-component requisite. Before you put this component into production, you **must** install service for the component indicated in the message.

If you see the following message in the apply message log:

```
    ===> VMFVIEW - Message Log Browse of $VMFAPP  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = n  <===> Number of messages not shown = nn
********************************************************************************
****    PPFNAME: ESA        COMPNAME: AVS         APPID: 2VMVMD40   ****
********************************************************************************
****            Date: mm/dd/yy            Time: hh:mm:ss            ****
********************************************************************************
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCL which is a
CK:            higher level than VVTVM.  The higher-level service may need
CK:            to be reworked or removed.
CK:VMFAPP2121I VVTLCL = xxxxxxx.xxxxxxxx
```

then rework local service.

Rework Local Service

If you have local modifications or APAR fixes (not in VMSES/E format), that have been affected by the applied service, for AVS, go to Chapter 7, “Reworking Local Service and Modifications” on page 7-1. Complete the procedures, and then return here.

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

1 If you have serviced VMSES/E, access the 5E6 disk to use the new level of VMSES/E. This will replace the access to the 5E5 disk which contains the older level of VMSES/E.

```
access 5e6 b
```

2 If your PPF file has changed since you performed the last task for this component step, enter the following command:

```
vmfppf esa avs|avssfs
```

3 If your access order has changed since you performed the last task in this component step, enter the following command:

```
vmfsetup esa avs|avssfs
```

Update the Build Status Table

1 Update the Build Status Table with serviced parts:

vmfbld ppf esa avs|avssfs (status

This command will update the Build Status Table with a status of **serviced** for each object that needs to be built.

2 If the $PPF files have been serviced then you get the following prompts and need to answer the prompts and enter the following commands. Otherwise, continue to 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
VMFBLD2185R 2VMVMD40 $PPF
VMFBLD2185R When source product parameter files are serviced, all product
            parameter files built from them must be recompiled using VMFPPF
            before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product parameter files
            built to your A-disk and exit VMFBLD so you can recompile your
            product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already recompiled your
            product parameter files with VMFPPF
```

0

Enter 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 2VMVMD40 $PPF on 191 (A) from level $PFnnnnn
```

vmfppf esa avs|avssfs

Choose between **avs|avssfs**, but do not enter the **|**. Use the *compname* **avs** or **avssfs** if you have a GCS/XA environment.

copy 2vmvmd40 $ppf a = = *fm-51d* (olddate replace
erase 2vmvmd40 $ppf a

Enter the VMFPPF command to ensure you have the appropriate level of PPF.

fm-51d is the file mode of the software inventory disk.

vmfsetup esa avs|avssfs

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf esa avs|avssfs (status
1

Enter the VMFBLD command to complete updating the build status table. When you receive the prompt that was previously displayed, enter 1 to continue.

3 Use VMFVIEW to review the build status messages, and see what objects need to be rebuilt:

vmfview build

Build Serviced Objects

Note

Use the remainder of this section to build all AVS objects that were serviced. If you prefer to build AVS objects individually, then use “Detailed Build Steps: AVS” on page 4-33. Then return to “Rebuild the HELP Files in Uppercase (Optional).”

1 Use the *compname* **avs** or **avssfs** if you have a GCS/XA environment.

vmfbld ppf esa avs|avssfs (serviced

Choose between **avs|avssfs**, but do not enter the **|**.

Note:

If you receive the following message:

```
VMFSBR2000I  Objects in segment build list
             fn ft have been built or deleted.
             Any segments using this build
             list will have to be rebuilt.
```

You must rebuild the indicated saved segment if you have installed it. This segment rebuild will be handled in “Step 11. Place Your Serviced Components into Production” on page 3-68.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the HELP Files in Uppercase (Optional)

If you want serviced HELP files to be uppercase (for example, for NLS purposes) on the MAINT 402 disk, enter the following commands:

vmfppf uceng avs|avssfs

This command will create the UCENG PPF file needed to put the HELP files in uppercase.

vmfbld ppf uceng avs|avssfs agwblhlp (serviced setup

This copies all the serviced HELP files to the 402 disk and puts them in uppercase.

If the $PPF file has been serviced, refer to “Update the Build Status Table” to help you answer the prompts. Remember when entering the commands to use the *ppfname* **uceng** rather than **esa**.

Merge Processing

1 Clear the alternate apply disk and move the new service to the intermediate apply disk. VMFMRDSK will copy the files to the intermediate apply disk and then erase them from the alternate apply disk. This ensures that service on the intermediate apply disk matches the test build disks (490 and 493).

After performing the merge you will no longer be able to remove the current service level as described in Chapter 8, “Removing Service” on page 8-1 since the current service level will be empty. Before doing the merge you need to be sure the service is tested to your satisfaction and that you are ready to commit it to this service level. For components that share disks, like AVS and TSAF, doing a merge on one component will merge all the files from the other component.

```
vmfmrdsk esa avs|avssfs apply (setup
```

2 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview merge
```

Where to Next?

Path you are following	Go to this section
If you are installing corrective service (COR), or PSP Bucket COR, by following Chapter 3, “Using the COR Service Procedure” on page 3-1.	• “Step 11. Place Your Serviced Components into Production” to put this component's new service into production. This should be done **only once** at the completion of all service application.
If you are installing service from the RSU tape, then go to	“Process the PSP Bucket COR Tape” on page 2-14 to receive service from the PSP bucket COR tape.

Step 11. Place Your Serviced Components into Production

This step assumes that you have performed any testing that you find necessary on the service that you have received and are ready to put it into production. It also assumes that you have serviced all components for this service session. In this procedure you will:

- Rebuild the saved segments
- Rebuild the NLS Segment (Optional)
- Move files into the byte file system
- Place the test build disks into production
- Rebuild the CMS saved system

- Place the CP nucleus and CMS into production
- Place TSAF and AVS into production
- Place GCS into production, and
- Merge your tested service to your production service disks.

Attention

This entire step will affect your production environment. All users must be logged off during this step.

Rebuild the Saved Segments

The following section provides information for rebuilding the CMS saved segments. You only need to rebuild a saved segment if you have it installed and it was serviced. In this section the VMFBLD command with the SERVICED option is used and is the recommended command. If for some reason you want to rebuild the segments even if they were not serviced, use the ALL option instead. For more information about saved segments, see *VM/ESA: Planning and Administration*.

1 The CMS saved segments are defined using the ESA PPF file. If you have your own override PPF, **you must use** the VMFSGMAP command to update the PPF name (and if you changed it, the component name) on the :BLDPARMS tag for each segment definition. See the *VM/ESA: VMSES/E Introduction and Reference* for instructions.

2 Log on MAINT.

Note

If you use a user ID other than MAINT to apply service, you must update the VMSESE PROFILE file and the :DCL section of the SEGBLD PPF file to change the owner of the 51D disk from MAINT to your user ID. For examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *VM/ESA: VMSES/E Introduction and Reference*.

3 Prepare to build the saved segments:

Attention

The commands in this step must be entered exactly as shown. The only variable is the number of megabytes of storage defined.

```
set machine xa
System reset.
System = XA
```

```
define storage nnm
STORAGE =      nnM
storage cleared - system reset
```

nn is the amount of megabytes of storage plus the storage used by CMS. Define your storage to be greater than the highest storage location of your segments.

ipl 190 clear parm nosprof instseg no	IPL CMS without executing the system profile (SYSPROF EXEC) (which, for example, loads the CMSPIPES saved segment) and without loading the installation saved segment (CMSINST).

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

***** DO NOT press ENTER! *****

access (noprof	Bypass the execution of the PROFILE EXEC.

4 Establish the minidisk access order and update the build status table:

access 5e6 b	Access the test VMSES/E installation and service tools disk.
access 51d d	Access the software inventory disk.
vmfppf segbld esasegs	**segbld** is a PPF and **esasegs** is a special component name for building segments.
vmfbld ppf segbld esasegs (status	Enter this command to determine which segments need to be built because they were serviced.
vmfview build	View the build message log.

Note

If you would like to build the individual segments one at a time, then skip this note and continue with 5 on page 3-71.

Otherwise, if you want to build all serviced CMS segments enter the following command and then go to 16 on page 3-72.

vmfbld list segbld esasegs cmssegs blddata (serviced

The following message might be received:

DMSDCS1083E SAVED SEGMENT $$DMY$$ does not exist (RC=44)

It is because either one of the CMSBAM segment generation execs issued the commands "SET SYSNAME CMSBAM $$DMY$$" and "SET DOS ON" or the CMSBAM segment was rebuilt. The message is normal and can be ignored. No action is required.

Note for VMFBLD PPF commands

If, for some reason, when you use the VMFBLD PPF command to build a segment and the build is unsuccessful you can reissue the command with the additional operand of **TYPE**. **TYPE** should be specified before the **(**. This will allow you to see the messages that are usually hidden by VMFBDSEG (the segment part handler). (Do not use this operand with the VMFBLD LIST commands.)

Also, do not put the following commands in an EXEC because it will be possible that some of the segment builds will not complete if CMS uses the same storage as where the segments reside.

5 Build the CMSFILES saved segment:

```
vmfbld ppf segbld esasegs segblist cmsfiles (serviced
```

6 Build the SVM saved segment:

```
vmfbld ppf segbld esasegs segblist svm (serviced
```

7 Build the CMSVMLIB logical saved segment:

```
vmfbld ppf segbld esasegs segblist cmsvmlib (serviced
```

8 Build the CMSPIPES saved segment:

```
vmfbld ppf segbld esasegs segblist cmspipes (serviced
```

9 Build the DOSINST saved segment:

```
vmfbld ppf segbld esasegs segblist dosinst (serviced
```

10 Build the CMSBAM and CMSDOS saved segments:

```
vmfbld list segbld esasegs dosbam blddata (serviced
```

Build with the LIST operand.

CMSBAM and CMSDOS belong to the DOSBAM space and are built together using the DOSBAM BLDDATA list.

The following message might be received:

```
DMSDCS1083E SAVED SEGMENT $$DMY$$ does not exist (RC=44)
```

It is because either one of the CMSBAM segment generation execs issued the commands "SET SYSNAME CMSBAM $$DMY$$" and "SET DOS ON" or the CMSBAM segment was rebuilt. The message is normal and can be ignored. No action is required.

11 Build the CMSINST logical saved segment:

```
vmfbld ppf segbld esasegs segblist instseg (serviced
```

12 Build the HELP logical saved segment:

```
vmfbld ppf segbld esasegs segblist helpseg (serviced
```

13 Build the GUICSLIB logical saved segment:

```
vmfbld ppf segbld esasegs segblist guicslib (serviced
```

14 Build the GUIVMGUI logical saved segment:

```
vmfbld ppf segbld esasegs segblist guivmgui (serviced
```

15 If necessary, correct any problems before going on. You might see other build requirements not relating to VM/ESA that will need to be resolved.

```
vmfview build
```

16 If you received the message:

```
VMFBLD2006E Build list bldlist in component compname
            of ppfname PPF could not be processed because
            its status was neither BUILT nor DELETED.
```

from any of the preceding builds, then you will need to issue the VMFBLD command to update the status.

```
vmfbld ppf ppfname compname bldlist  (serviced setup
```

ppfname, *compname* and *bldlist* names were specified in the message.

17 If you received the message:

```
VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and
            must be moved to the S disk.
```

from any of the preceding builds, then update the SYSTEM SEGID on the 490 disk. The 490 disk will be put into production as the 190 disk on “Place the Test Build Disks into Production” on page 3-74.

Attention

This **must** be done now or it will cause an error later.

Do not IPL until you put the 490 into production.

access 490 t
copyfile system segid *fm-51d* **= = t2 (replace olddate**

Copy with a file mode of **2**.

If there are other users logged on (even though this is not recommended when you began this step), you might want to copy the SYSTEM SEGID file to the 190 disk now instead of waiting until the 490 disk is put into production. This will eliminate the possible opportunity for another user to IPL and program check. COPY with a file mode of **2**.

access 190 t
copyfile system segid *fm-51d* **= = t2 (replace olddate**

Rebuild the NLS Segment (Optional)

The following table identifies the name for each national language saved segment. If you have an NLS segment, enter the VMFBLD command with the *segname* for the national language desired.

Table 3-1. National Language Segments

segname	**National Language**
NLSKANJI	Kanji
NLSUCENG	Uppercase English
NLSFRANC	French
NLSGER	German
NLSCANFR	Canadian French

vmfbld ppf segbld esasegs segblist *segname* **(serviced**

Move Files into the Byte File System

The following section provides information on moving Java and NetRexx files into the byte file system after they have been serviced. You only need to move these files if you have installed Java and NetRexx in the byte file system. In this section, the VMFBLD command with the SERVICED option is used and is the recommended command. If for some reason you want to move the files and they were not serviced, use the ALL option (not the SERVICED option).

1 Establish the minidisk access order and update the build status table.

```
access 5e6 b
```

```
vmfppf esa cmsprod
vmfsetup esa cmsprod
vmfbld ppf esa cmsprod (status
vmfview build
```

cmsprod is a special component name used for CMS production steps.

2 Move the Java files to the byte file system.

```
vmfbld ppf esa cmsprod dmsbljav (serviced
```

3 Move the NetRexx files to the byte file system.

```
vmfbld ppf esa cmsprod dmsblnrx (serviced
```

4 If necessary, correct any problems before going on.

```
vmfview build
```

Place the Test Build Disks into Production

Note

If you have updated the 190 or 193 disks you should copy any new and changed files to the 490 and 493 disks. This will avoid accidentally back-leveling modified files.

1 Use the COPYFILE command to copy the test VMSES/E tools disk (5E6) to the production VMSES/E tools disk (5E5).

Note: Use your installation back-up procedures if you want to save a back-up of the 5E5 disk. You need to perform the COPYFILE immediately after the ACCESS so that you do not inadvertently re-access the disk and restore the files.

```
access 5e6 t
access 5e5 b (erase
copyfile * * t = = b (olddate
release t
```

2 Use the COPYFILE command to copy the test system tools disk (493) to the production system tools disk (193).

Note: Use your installation back-up procedures if you want to save a back-up copy of the 193 disk. You need to perform the COPYFILE immediately after the ACCESS so that you do not inadvertently re-access the disk and restore the files.

```
access 493 l
access 193 m (erase
copyfile * * l = = m (olddate
release l
```

3 Use the DASD dump/restore program to copy the test CMS system disk (490) to the production CMS system disk (190). DDR must be used instead of COPYFILE because the nucleus will be copied from the 490 to the 190 disk.

Note: Use your installation back-up procedures if you want to save a back-up copy of the 190 disk.

```
ddr
VM/ENTERPRISE SYSTEMS ARCHITECTURE DASD DUMP/RESTORE PROGRAM
ENTER:
```

The system prompts you to enter a DDR control statement.

```
sysprint cons
```

Route messages to the console (rather than a printer).

```
ENTER:
input 490 dasd mnt490
```

Enter 490 as the source minidisk for the DDR.

490 is the virtual device number of the test CMS system disk. **dasd** will determine the device type of the real DASD volume on which the 490 minidisk resides (for example, 3380). **mnt490** is the label of the minidisk.

```
ENTER:
output 190 dasd mnt190
```

Enter 190 as the target minidisk for the DDR.

190 is the virtual device number of the production CMS system disk you are creating. **dasd** will determine the device type of the real DASD volume on which the 190 minidisk resides. **190** must be the same DASD type as the 490 disk. **mnt190** is the label of the 190 minidisk.

```
ENTER:
copy all
```

Enter the COPY control statement to copy the entire 490 minidisk.

```
END OF COPY
ENTER:
```

DDR copies and displays messages about the operation. When the processing is complete, DDR prompts you to enter another control statement.

ENTER

At the prompt, press ENTER to enter a null line.

```
END OF JOB
```

The system reports end-of-job status.

```
access 190 r
format 190 r (label
DMSFOR605R Enter disk label:
mnt190
```

Change the label of the 190 disk.

Access 190 read/write. You **must** use the LABEL option. If you do not, you will erase all of the files on the 190 minidisk.

Rebuild the CMS Saved System

1 Prepare to build the CMS saved system:

```
access 193 m
sampnss cms
```

Define a skeleton CMS saved system.

2 Build the CMS saved system

```
ipl 190 clear parm savesys cms
VM/ESA V2.4.0 mm/dd/yy hh:mm
```

Build the CMS saved system.

ENTER

Press ENTER to complete the IPL.

Place the CP Nucleus into Production

1 Release the primary parm disk from CP, link the parm disk write mode, and access the disk. Rename the current CPLOAD MODULE to *fn* MODULE. *fn* will be the file name that you used when you created the original backup copy during the installation of VM/ESA. Then copy the new CPLOAD MODULE, which is your CP nucleus, to the primary parm disk CF1. Enter the following commands:

```
query cpdisk
LABEL   USERID  VDEV MODE STAT VOL-ID RDEV  TYPE  STARTLOC  ENDLOC
MNTCF1  MAINT   0CF1  A   R/O  ES2RES rdev  ECKD        nn nn
MNTCF2  MAINT   0CF2  B   R/O  ES2RES rdev2 ECKD        nn nn
cprelease a

CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A
completed. Ready; T=0.01/0.01 13:49:06
```

When you do the CPRELEASE, CP loses access to the LOGO files on the CF1 parm disk. If you do not have backup copies of your LOGO files on the secondary parm disk, CF2, then HCPBOX will come up for any new LOGONs.

Also when you do the CPRELEASE, CP loses access to any log message files on the CF1 parm disk. If you do not have backup copies of your log message files on the secondary parm disk, CF2, the users will receive no log message at all.

```
link maint cf1 cf1 m

access cf1 u
access 493 t
```

```
erase cplold module u
```

Erase the backup CP nucleus.

Note: CPLOLD module is the default name used for the backup copy of the CP nucleus during installation.

```
rename cpload module u cplold module u
```

Rename the current CP nucleus to the backup CP nucleus.

```
copyfile cpload module t cpload module u (olddate replace
```

Copy the new CP nucleus to the parm disk.

2 Place the CP nucleus into production by shutting down and re-IPLing your system. Enter the following command:

```
shutdown reipl
```

Note

If the IPL failed and you are IPLing second level, re-IPL with the Back-Up nucleus on the CF1 parm disk. If you are IPLing first level, see the appropriate (ES/3090™, ES/4381™ or ES/9000®) *Processor Complex Operator's Guide for the System Console* for instructions.

ipl *rdev* **clear loadparm** *cons*	*rdev* is the address of the real DASD device containing your system residence volume. *cons* is the address of your console Change the module name from CPLOAD to CPLOLD. Press the PF10 key to load the CP module that is contained on the CF1 parm disk. For more information see *VM/ESA: Planning and Administration*.

If you could not use the CF1 parm disk, repeat this process with the CF2 parm disk.

Place CMS GUI into Production

If the workstation agents have been serviced (build list GUIBLAGT), the enduser must obtain access to the new workstation agent (refer to *IBM VM/ESA: Graphical User Interface Facility*.

Place TSAF and AVS into Production

TSAF and AVS production level code resides on the 193 disk. Make sure all users of this code have re-linked and re-accessed the MAINT 193 disk.

Copy the PROFTSAF EXEC from the MAINT 193 disk to the TSAFVM 191 disk as PROFILE EXEC. This step is only required if PROFTSAF EXEC has been serviced.

Place GCS into Production

Make sure you re-IPL your GCS users.

Merge Tested Service Level

This section should be done after the new code has been performing to your satisfaction for a few days (or few weeks). After merging the service in this section with the VMFMRDSK commands you will not be able to remove the service.

In this procedure you will be merging your alternate and intermediate (test) disks to your production disks. The alternate and intermediate disks represent your tested service that you just placed into production. This service is now ready to be committed prior to additional service. Complete the following commands to merge the service to your production disks for every component.

Note: You may elect not to merge the service at this time and merge it prior to installing additional service. This procedure then would become a preparation step for your installation.

Table 3-2. Component Name

compname	*prodid* **and default** *appid* **Identifier**
VMSES	2VMVMK40
REXX	2VMVMF40
CMS	2VMVMA40
CP	2VMVMB40
DV	2VMVMI40
GCS	2VMVML40
TSAF	2VMVMH40
AVS	2VMVMD40
GCSSFS	2VMVML40
TSAFSFS	2VMVMH40
AVSSFS	2VMVMD40

1 Access the necessary VMSES/E disks:

```
access 5e5 b
access 51d d
```

2 If your PPF file has changed since you performed the last task in this component step, refer to Table 3-2, then enter the following command:

vmfppf esa *compname*

3 Enter the following command:

vmfsetup esa *compname*

4 Merge the files from the alternate and intermediate apply disks to the apply production disks for each component. The values for *compname* are listed in Table 3-2.

vmfmrdsk esa *compname* **apply (levels 2**
vmfmrdsk esa *compname* **apply (levels 2**

You need to enter VMFMRDSK twice to ensure that all service levels are merged to the apply production disk. Since there are three disks-alternate, intermediate and production- you need to issue the command twice. The first command merges the intermediate disk to the production disk and clears the alternate disk by merging to the intermediate disk. The second command clears the intermediate disk by merging the files to the production disk.

Create a New Back-Up Copy of the Primary Parm Disk

After your system has been performing to your satisfaction for a few days (or few weeks), create a backup copy of your primary parm disk.

1 Link and access the CF2 parm disk in write mode.

cprelease b

link maint cf2 cf2 m

access cf2 *fm2*

fm2 is any free file mode.

2 Use the ACCESSM0 command to allow files with a file mode of “0” to be copied. Then copy all of the files from the CF1 parm disk to the CF2 parm disk.

access 193 r

accessm0 on

access cf1 *fm1* **(mode0**

fm1 is any free file mode.

copyfile * * *fm1* **= =** *fm2* **(olddate replace**

fm2 is the file mode that you used when accessing the CF2 parm disk.

3 Reset your environment.

release *fm2* **(detach**
cpaccess maint cf2 b rr
release *fm1*
accessm0 off

Note

You have now finished servicing VM.

Chapter 4. Detailed Component Build Steps

This chapter is supplemental to Chapter 3, Using the COR Service Procedure. It contains the detailed build procedures for each of the components of VM/ESA. It is included to give you the option of building objects one build list at a time. The steps are separated by component. Go directly to the component you are servicing and use the tasks within that section.

Detailed Build Steps: VMSES/E

Determine Which Build Lists Need to be Processed

1 Determine which build steps need to be performed:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-1, and then run the corresponding build steps in the order that they appear in the table.

Table 4-1. VMSES/E Build Lists

Build List	Build Procedure
VMFBLHLP VMFBLNON VMFBLSES VMFBLSRC VMFBLSYS	"Select and Copy Serviced Files"
VMFMLOAD VMFSLOAD	"Rebuild the Executable MODULEs" on page 4-2
VMFSBHLP	"Process the Segment Build List" on page 4-3

Select and Copy Serviced Files

Table 4-2 on page 4-2 specifies build lists for VMSES/E that require the select and copy build functions.

Table 4-2. Build Lists for VMSES/E Select and Copy Functions		
Build List	**Description**	**Test/Production Disk**
VMFBLHLP	Build list for HELP files.	19D/19D
VMFBLNON	Build list for non-VMSES/E files on the S-disk	490/190
VMFBLSES	Build list for the 5E6 disk	5E6/5E5
VMFBLSRC	Build list for source files.	493/193
VMFBLSYS	Build list for the S-disk	490/190

1 Enter the following command for each build list in Table 4-2 that was in the $VMFBLD $MSGLOG.

vmfbld ppf esa vmses *bldlist* **(serviced**

bldlist is the file name of one of the build lists in Table 4-2.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Executable MODULEs

Table 4-3 describes the build lists used to rebuild executable MODULEs.

Table 4-3. Build Lists for VMSES/E Executable MODULEs		
Build List	**Description**	**Test/Production Disk**
VMFMLOAD	Build list for VMSES/E MODULEs that reside on the 5E5 disk.	5E6/5E5
VMFSLOAD	Build list for VMSES/E MODULEs that reside on the system disk.	490/190

For each of the build lists in Table 4-3, complete these tasks:

1 If any of the files in the build list have been serviced, enter the VMFBLD command to rebuild all of the serviced MODULEs in that build list:

vmfbld ppf esa vmses *bldlist* **(serviced**

bldlist is the file name of one of the build lists in Table 4-3.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes* or use online HELP.

```
vmfview build
```

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-4. Build List for VMSES/E Segment Build List

Build List	Description
VMFSBHLP	Build list for VMSES/E files that reside on the HELP segment

1 Enter the following command to process the HELP segment:

```
vmfbld ppf esa vmses vmfsbhlp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Access the New Level of VMSES/E

To use the new serviced level of VMSES/E, enter the following command:

```
access 5e6 b
```

Note

You have now finished the detailed build steps for VMSES/E. Go to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-12 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: REXX/VM

Determine Which Build Lists Need to be Processed

1 To determine which build steps to perform, enter the following command:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-5, and then run the corresponding build steps in the order that they appear in the table.

Table 4-5. REXX/VM Build Lists

Build List	Build Procedure
IXXBLSRC	"Select and Copy Serviced Files"
IXXOM	"Update the Macro Libraries (MACLIBs)"
IXXBLNUC	"Rebuild the Nuclei" on page 4-5

Select and Copy Serviced Files

Table 4-6 contains a build list for REXX/VM that requires the select and copy build functions.

Table 4-6. Build List for REXX/VM Select and Copy Functions

Build List	Description	Test/Production Disk
IXXBLSRC	Build list for new/replacement source files.	493/193

1 Enter the following commands for the build list in Table 4-6 that was in the $VMFBLD $MSGLOG.

```
vmfbld ppf esa rexx ixxblsrc (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Update the Macro Libraries (MACLIBs)

Note

If you already updated the MACLIBs as part of reworking local service earlier in this component step, skip this step, and go to "Rebuild the Nuclei" on page 4-5.

Table 4-7 lists the REXX/VM MACLIB that may need to be rebuilt.

Table 4-7. Build List for REXX/VM Maclib

Build List	Description	Test/Production Disk
IXXOM	Build list for the IXXOM MACLIB.	490/190

1 Enter the VMFBLD command to rebuild the IXXOM MACLIB. When servicing a MACLIB using VMSES/E, ensure that your A-disk has enough free space to hold two copies of the MACLIB. This is to allow for the staging of the MACLIB, updating the serviced MACRO and COPY files and for the MACLIB compression.

```
vmfbld ppf esa rexx ixxom (serviced
```

ixxom is the name of the MACLIB from Table 4-7 on page 4-4.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Nuclei

Note

Parts of the REXX/VM component are contained in the CMS nucleus and the GCS nucleus. You must receive, apply, and build any REXX/VM service before you build either nucleus. If you have applied REXX/VM service and received message VMFBDU2180I to rebuild CMS and GCS nuclei, then you must rebuild and IPL the CMS nucleus and, if you have installed GCS, the GCS nucleus. If you are going to service CMS or GCS, then do not rebuild that nucleus now because it will be rebuilt when that component is serviced. If there is no service to CMS or GCS, then rebuild and IPL its nucleus now. For information on rebuilding and IPLing the CMS nucleus, see “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15. For information on rebuilding and IPLing the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27.

Note

You have now finished the detailed build steps for REXX/VM. Go to “Merge Processing” on page 3-19 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: CMS

Determine Which Build Lists Need to be Processed

1 To determine which build steps you should perform:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-8, and then run the corresponding build steps in the order that they appear in the table.

Table 4-8 (Page 1 of 2). CMS Build Lists

Build List	Build Procedure
DMSBL400 DMSBL490 DMSBL493 DMSBLASM DMSBLCRX DMSBLDOS DMSBLHLP DMSBLINF DMSBLNUH DMSBLRAS DMSBLSRC GUIBLEXS	"Select and Copy Serviced Files" on page 4-7
AMQOM DMSGPI DMSOM FPLGPI FPLOM MVSXA OSMACRO OSMACRO1 OSPSI	"Update the Macro Libraries (MACLIBs)" on page 4-8
CMSSAA DMSBLAEN DMSBLAMT DMSBLBKW DMSBLCEN DMSBLCMT	"Rebuild the Text Libraries" on page 4-9
DMSBLPRP	"Rebuild the Load Library" on page 4-9
CMSMLOAD CMSXLOAD IOCPLOAD SERVLOAD	"Rebuild the Executable MODULEs" on page 4-10
DMSBLFAS	"Rebuild the Assemble Module" on page 4-11
DMSBLVMT DMSBLVML GUIBLCSL	"Rebuild the Callable Services Libraries (CSLs)" on page 4-11

Table 4-8 (Page 2 of 2). CMS Build Lists	
Build List	**Build Procedure**
DMSSBBAM DMSSBCAN DMSSBDOS DMSSBFRA DMSSBGER DMSSBHLP DMSSBINS DMSSBKAN DMSSBPIP DMSSBRTS DMSSBSFS DMSSBSVM DMSSBUCE DMSSBVML GUISBCSL GUISBDSK	“Process the Segment Build Lists” on page 4-12
CMSLOAD	“Rebuild the CMS Nucleus” on page 4-13

Select and Copy Serviced Files

Table 4-9 specifies build lists for CMS that require the select and copy build functions.

Table 4-9. Build Lists for CMS Select and Copy Functions		
Build List	**Description**	**Test/Production Disk**
DMSBL400	Build list for files for the 400 disk.	400/400
DMSBL490	Build list for EXEC, XEDIT and other files for the 490/190 disks.	490/190
DMSBL493	Build list for EXEC, XEDIT and other files for the 493/193 disks.	493/193
DMSBLASM	Build list for the F assembler text decks.	493/193
DMSBLCRX	Build list for the compiled REXX/VM files.	490/190
DMSBLDOS	Build list for CMS text decks used by CMSDOS and CMSBAM.	493/193
DMSBLHLP	Build list for HELP files.	19D/19D
DMSBLINF	Build list for CMS/Dump Viewing Facility interface files.	493/193
DMSBLNUH	Build list for HELP files that cannot be uppercased.	19D/19D
DMSBLRAS	Build list for sample RAS tools.	493/193
DMSBLSRC	Build list for new/replacement source files.	493/193
GUIBLEXS	Build list for GUI mods segment files.	490/190

1 Enter the following command for each build list in Table 4-9 that was identified in the $VMFBLD $MSGLOG.

vmfbld ppf esa cms *bldlist* **(serviced** *bldlist* is the build list in Table 4-9.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Update the Macro Libraries (MACLIBs)

Note

If you already updated the MACLIBs as part of reworking local service earlier in this component step, skip this step, and go to "Rebuild the Text Libraries" on page 4-9.

Table 4-10 contains a list of CMS MACLIBs that may need to be rebuilt.

Table 4-10. Build Lists for CMS MACLIBs

Build Lists	Description	Test/Production Disk
AMQOM	Build list for the AMQOM MACLIB.	490/190
DMSGPI	Build list for the DMSGPI MACLIB.	490/190
DMSOM	Build list for the DMSOM MACLIB.	490/190
FPLGPI	Build list for the FPLGPI MACLIB.	490/190
FPLOM	Build list for the FPLOM MACLIB.	490/190
MVSXA	Build list for the MVSXA MACLIB.	490/190
OSMACRO	Build list for the OSMACRO MACLIB.	490/190
OSMACRO1	Build list for the OSMACRO1 MACLIB.	490/190
OSPSI	Build list for the OSPSI MACLIB.	490/190

1 Enter the VMFBLD command to rebuild the MACLIB for each build list that needs to be rebuilt. When servicing a MACLIB using VMSES/E, ensure that your A-disk has enough free space to hold two copies of the MACLIB. This is to allow for the staging of the MACLIB, updating the serviced MACRO and COPY files and for the MACLIB compression.

vmfbld ppf esa cms *bldlist* **(serviced** *bldlist* is the name of the build list Table 4-10.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Text Libraries

Table 4-11 describes the build lists used to rebuild the CMS TXTLIBs.

Table 4-11. Build Lists for CMS TXTLIBs

Build Lists	Description	Test/Production Disk
CMSSAA	Build files for the CMSSAA TXTLIB.	490/190
DMSBLAEN	Build files for the DMSAENV TXTLIB.	490/190
DMSBLAMT	Build files for the DMSAMT TXTLIB.	490/190
DMSBLBKW	Build files for the BKWLIB TXTLIB.	493/193
DMSBLCEN	Build files for the DMSCENV TXTLIB.	490/190
DMSBLCMT	Build files for the DMSCMT TXTLIB.	490/190

1 For each build list in Table 4-11 that has been serviced, enter the following command to create the updated level of the TXTLIB associated with it.

```
vmfbld ppf esa cms bldlist (serviced
```

bldlist is the build list in Table 4-11.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Load Library

Table 4-12 contains the load library that may need to be rebuilt.

Table 4-12. Build List for CMS Load Library

Build List	Description	Test/Production Disk
DMSBLPRP	Build list for the PROPLIB LOADLIB.	490/190

1 Enter the VMFBLD command to rebuild the load library for each build list that needs to be rebuilt.

```
vmfbld ppf esa cms dmsblprp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Executable MODULEs

Table 4-13 describes the build lists used to rebuild executable MODULEs.

Table 4-13. Build Lists for CMS Executable MODULEs

Build List	Description	Test/ Production Disk
CMSMLOAD	Build list for CMS MODULEs that reside on the system disk.	490/190
CMSXLOAD	Build list for CMS MODULEs that reside on the system tools disk.	493/193
IOCPLOAD	Build list for IOCP MODULEs that reside on the system disk.	490/190
SERVLOAD	Build list for SFS server MODULEs that reside on the BASE disk.	493/193

1 For each build list in Table 4-13 that has MODULEs that have been serviced, enter the VMFBLD command to rebuild the serviced modules:

Note: If you have DB2 Server for VM or ISPF or both, and are using the CMS Pipelines interfaces to these applications you must enter the GLOBAL command to identify the text libraries containing the interface files before running VMFBLD.

```
global txtlib pipsql
```

pipsql is the file name of the TXTLIB containing the DB2 Server for VM interface files, ISPF interface files, or both.

vmfbld ppf esa cms *bldlist* **(serviced**

bldlist is the file name of one of the build lists from Table 4-13.

After each VMFBLD command, review the build message log (as described in the next step) before you proceed to the next.

Notes:

a. If you receive the following messages:

```
DMSLIO201W The following names are undefined: ISPLINK ARIPRDI
```

disregard it, because they are interface files for ISPF and SQL (if you are not using these as interfaces into DMSPIPE).

b. If, while building the DMSCUT or DMSDFT MODULEs, you get a return code of 4 with this message:

```
DMSLIO201W The following names are undefined: DMSUSRX1
DMSUSRX2
```

disregard it, because the undefined names are user exits.

c. If, while building the DMSPIPE MODULE, you get a return code of 4 with this message:

```
DMSLIO201W The following name is undefined: DMSDSCSC
```

disregard it, because the undefined name is a user exit.

d. If you receive the following message while building the CSL libraries:

```
DMSSOP2534I No MACLIB libraries have been globaled.
```

you can disregard it.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Assemble Module

Table 4-14 contains a CMS build list and MODULE that may need to be rebuilt, which requires special processing.

Table 4-14. Build List for CMS MODULEs

Build List	Description	Test/Production Disk
DMSBLFAS	ASSEMBLE MODULE	490/190

1 Enter the VMFBLD command for the build list in Table 4-14 to rebuild the F assembler if any of the requisite files have been serviced.

```
vmfbld ppf esa cms dmsblfas (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Callable Services Libraries (CSLs)

Table 4-15 on page 4-12 describes the build lists used to rebuild VMMTLIB and VMLIB.

Table 4-15. Build Lists for CMS CSL Libraries

Build Lists	Description	Test/Production Disk
DMSBLVMT	Build list for VMMTLIB	490/190
DMSBLVML	Build list for VMLIB	490/190
GUIBLCSL	Build list for VMGUILIB	490/190

1 The following command will build the VMLIB CSLLIB, VMLIB CSLSEG, VMGUILIB CSLSEG , and the VMMTLIB TEXT if any of their parts have been serviced.

vmfbld ppf esa cms *bldlist* **(serviced**

bldlist is the name of the CSL build list from Table 4-15.

Note: If you receive the following message while building the CSL libraries:

```
DMSSOP2534I No MACLIB libraries have been
globaled.
```

you can disregard it.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Process the Segment Build Lists

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-16. Build Lists for CMS Segment Build Lists	
Build List	**Description**
DMSSBBAM	Build list for the CMSBAM saved segment
DMSSBCAN	Build list for national language Canadian French
DMSSBDOS	Build list for the CMSDOS saved segment
DMSSBFRA	Build list for national language French
DMSSBGER	Build list for national language German
DMSSBHLP	Build list for CMS files that reside with the HELP segment
DMSSBINS	Build list for the CMSINST saved segment
DMSSBKAN	Build list for national language Kanji
DMSSBPIP	Build list for the CMSPIPES saved segment
DMSSBRTS	Build list for DMSRTSEG logical segment files that reside with CMSVMLIB
DMSSBSFS	Build list for the CMSFILES saved segment
DMSSBSVM	Build list for the SVM saved segment
DMSSBUCE	Build list for national language Uppercase English
DMSSBVML	Build list for VMLIB logical segment files that reside with CMSVMLIB
GUISBCSL	Build list for GUICSLIB saved segment
GUISBDSK	Build list for GUIVMGUI saved segment

1 Enter the following command for each segment build list, if it has been serviced:

vmfbld ppf esa cms *bldlist* **(serviced**

bldlist is a build list from Table 4-16.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

Rebuild the CMS Nucleus

Note

Parts of the REXX/VM component are contained in the CMS nucleus and the GCS nucleus. You must receive, apply and build any REXX/VM service before you build either nucleus. If you have applied REXX/VM service, and there is no service for CMS or GCS, you must still rebuild and IPL the CMS nucleus and, if you have installed GCS, the GCS nucleus.

Note for NLS

If you would like to change the default system national language for CMS you must rebuild and IPL the CMS nucleus to contain the language. To do this, follow the steps in Appendix C, “Building the NLS Nuclei, Including UCENG” on page C-1 to change the PPF file if necessary. Then continue with “Rebuild the CMS Nucleus” and “IPL the CMS Nucleus” on page 4-15. In these procedures substitute the national language $PPF file name for *ppfname*.

1 Make sure your machine environment is correct (XA), your storage is defined to at least 35M and the necessary disks are accessed. If your environment is not correct follow the instructions on “Step 1. Prepare the System for Service” on page 3-5 (steps 3 and 4).

2 If you make any changes to the DMSNGP ASSEMBLE file, you must reassemble the file.

vmf-assemble **dmsngp esa cms ($select outmode localmod logmod**

vmf-assemble can be the VMFASM, VMFHASM, or VMFHLASM command depending on which assembler you have on your system.

Using these options, DMSNGP TXT will be placed into your $SELECT file. The DMSNGP *modid* file will also be placed on your LOCALMOD disk and the local VVT will be updated if necessary.

For more information about the DMSNGP ASSEMBLE file, see the *VM/ESA: Planning and Administration* book.

3 Build the nucleus.

spool prt *
spool pun *

Spool the printer and punch output to your virtual reader.

vmfbld ppf esa cms cmsload (all setup

cmsload is the build list identified by the VMFBDNUC entry in the :BLD section of the component PPF, unless you use your own override file.

```
LOAD LIST:  $$$TLL$$ EXEC     A1 mm/dd/yy hh:mm (MNT191)
RDR FILE 0025 SENT FROM MAINT   PUN WAS 0025 RECS 043K CPY  001 A NOHOLD NOKEEP
```

The punch file containing the nucleus is spooled to your reader.

4 Check the results of the build. Use VMFVIEW to examine the $VMFBLD $MSGLOG file to verify that the correct levels of the text decks were loaded into the nucleus.

```
vmfview build
```

IPL the CMS Nucleus

1 Verify that the new nucleus is in your reader. Note the spool ID for use in later commands.

```
query rdr * all
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME      TYPE     DIST
MAINT    0025 A PUN 00043417 001 NONE 05/30 14:39:14 $$$TLL$$  IPL      SYSPROG
```

2 Make the file that contains the new nucleus the first file in your reader. If you have two copies order the reader so that the most recent one is first.

order rdr *spoolid*

spoolid is the spool ID of the nucleus file that you recorded earlier. In the previous example, the spool ID was 0025.

3 IPL the new nucleus.

define 190 590

Ensure that the nucleus gets loaded to the CMS test system disk, and does not overlay your production nucleus.

define 490 190

Note

The 490 disk is defined as 190 because the default address in DMSNGP for the nucleus is 190. If you have changed DMSNGP, substitute your address for the 190 in the two commands.

ipl 00c clear

Note: At this point, you might receive one or more nucleus generation prompts. A prompt appears if a **?** appears as an entry in, or if information is missing from the DMSNGP ASSEMBLE file. For more information about how to respond to these prompts, see the discussion of the DMSNGP profile in the chapter “Tailoring CMS with the DEFNUC macro” in the *VM/ESA: Planning and Administration* manual.

```
HCPNSS446E The Named Saved System (NSS) CMS was not
           previously defined and cannot be saved.
```

If you changed the SAVESYS operand in DMSNGP ASSEMBLE to SAVESYS=YES you will receive message HCPNSS446E. You can ignore this message. The CMS named saved system will be saved in “Step 11. Place Your Serviced Components into Production” on page 3-68.

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

If you defined your own version ID, it appears here and each time you IPL the CMS test system disk (MAINT 490).

ENTER

Press ENTER to complete the IPL.

4 Save the CMS load map.

access 493 *fm-493*

fm-493 is any free file mode.

spool prt nohold
close prt
RDR FILE 0026 SENT FROM MAINT PRT
WAS 0026 RECS 8270 CPY 001 A NOHOLD NOKEEP

The CMS load map has been spooled to your virtual reader. Note the spool ID for use in the next command.

receive *spoolid* **cmsnuc map** *fm-493* **(replace**

Receive the file onto the system tools disk, using the spool ID from the previous message. In the previous example, the spool ID was 0026.

vmfcopy cmsnuc map *fm-493* **= = = (prodid 2vmvma40%cms replace pack**

5 XEDIT the CMSNUC MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild and IPL the nucleus, see “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15. Then return to “Test the New Level of CMS” on page 3-30.

xedit cmsnuc map
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found

This response means that no unresolved symbols were found.

====> all/undefin/
DMSXDC546E Target not found

This response means that no undefined symbols were found.

====> qquit

6 Redefine the CMS disks back to their original values:

```
define 190 490
define 590 190
ipl 190 clear
VM/ESA V2.4.0 mm/dd/yy hh:mm
```

7 Access the necessary VMSES/E disks:

```
access address b
access 51d d
```

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

Note: You have now finished the detailed build steps for CMS. Go to "Test the New Level of CMS" on page 3-30 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: CP

Determine Which Build Lists Need to be Processed

1 To determine which build steps to perform, enter the following command:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-17, and then run the corresponding build steps in the order that they appear in the table.

Table 4-17. CP Build Lists	
Build List	**Build Procedure**
HCPBL490 HCPBL493 HCPBLHLP HCPBLIMG HCPBLINS HCPBLRAS HCPBLSAM HCPBLSDK HCPBLSRC	“Select and Copy Serviced Files” on page 4-18
HCPGPI HCPOM1 HCPOM2 HCPPSI	“Update the Macro Libraries” on page 4-19
HCPMLOAD HCPXLOAD	“Rebuild the Executable MODULEs” on page 4-20
HCPSBCAN HCPSBFRA HCPSBGER HCPSBHLP HCPSBKAN HCPSBUCE	“Process the Segment Build Lists” on page 4-20
HCPBLUTL	“Rebuild the CP Utilities” on page 4-21
CPLOAD	“Rebuild the CP Nucleus” on page 4-21

Select and Copy Serviced Files

Table 4-18 contains build lists for CP that require the select and copy build functions.

Table 4-18. Build Lists for CP Select and Copy Functions		
Build List	**Description**	**Test/Production Disk**
HCPBL490	Build list for EXEC, XEDIT and other files for the 490/190 disks.	490/190
HCPBL493	Build list for EXEC, XEDIT and other files for the 493/193 disks.	493/193
HCPBLHLP	Build list for HELP files.	19D/19D
HCPBLIMG	Build list for image library files.	493/193
HCPBLINS	Build list for install tools.	493/193
HCPBLRAS	Build list for CP dump tool files.	493/193
HCPBLSAM	Build list for sample files.	493/193
HCPBLSDK	Build list for text decks residing on the system disk.	490/190
HCPBLSRC	Build list for new/replacement source files.	493/193

1 Enter the following command for each build list in Table 4-18 that was in the $VMFBLD $MSGLOG.

vmfbld ppf esa cp *bldlist* **(serviced**

bldlist is the name of a build list in Table 4-18 on page 4-18.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Update the Macro Libraries

Note: If you already updated the MACLIBs as part of reworking local service earlier in this component step, skip this step, and go to "Rebuild the Executable MODULEs" on page 4-20.

Table 4-19 lists the CP MACLIBs that may need to be rebuilt.

Table 4-19. Build Lists for CP MACLIBs

Build Lists	Description	Test/Production Disk
HCPGPI	Build list for the HCPGPI MACLIB.	490/190
HCPPSI	Build list for the HCPPSI MACLIB.	490/190
HCPOM1	Build list for the HCPOM1 MACLIB.	493/193
HCPOM2	Build list for the HCPOM2 MACLIB.	493/193

1 Enter the VMFBLD command to rebuild the MACLIB for each build list that needs to be rebuilt. When servicing a MACLIB using VMSES/E, ensure that your A-disk has enough free space to hold two copies of the MACLIB. This is to allow for the staging of the MACLIB, updating the serviced MACRO and COPY files and for the MACLIB compression.

vmfbld ppf esa cp *bldlist* **(serviced**

bldlist is the name of a build list in Table 4-19.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Executable MODULEs

Table 4-20 describes the build lists used to rebuild executable MODULEs.

Table 4-20. Build Lists for CP Executable MODULEs

Build List	Description	Test/Production Disk
HCPMLOAD	Build list for VMFBLD to build CP MODULEs on the system disk.	490/190
HCPXLOAD	Build list for VMFBLD to build CP MODULEs on the system tools disk.	493/193

1 For each build list in Table 4-20 that has MODULEs that have been serviced, enter the VMFBLD command to rebuild the serviced MODULEs:

vmfbld ppf esa cp *bldlist* **(serviced**

bldlist is the file name of one of the build lists from Table 4-20.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Process the Segment Build Lists

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-21. Build Lists for CP Segment Build Lists

Build List	Description
HCPSBCAN	Build list for national language Canadian French
HCPSBFRA	Build list for national language French
HCPSBGER	Build list for national language German
HCPSBHLP	Build list for CP files that reside on the HELP segment
HCPSBKAN	Build list for national language Kanji
HCPSBUCE	Build list for national language Uppercase English

1 Enter the following command for each segment build list, if it has been serviced:

vmfbld ppf esa cp *bldlist* **(serviced**

bldlist is a build list from Table 4-21.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the CP Utilities

Table 4-22 shows the build list for CP utilities.

Table 4-22. Build List for CP Utilities

Build List	Description	Test/Production Disk
HCPBLUTL	Build list for CP Utilities.	490/190

1 Enter the following command to rebuild the serviced objects contained in HCPBLUTL.

```
vmfbld ppf esa cp hcpblutl (serviced
```

2 Use VMFVIEW to see if the object SA.DUMP was serviced and requires additional build steps.

```
vmfview build
BD:VMFBDU2180I HCPBLUTL  SA.DUMP    RUN HCPSADMP EXEC
```

If this message is shown then you need to rebuild the stand-alone dump.

3 If necessary, use the HCPSADMP EXEC to rebuild the Stand-Alone Dump and then re-IPL CMS.

```
hcpsadmp
ipl 190 clear
VM/ESA V2.4.0 mm/dd/yy hh:mm
```

HCPSADMP loads a Disabled Wait State PSW when it completes. For more information about this exec, see *VM/ESA: Planning and Administration*.

ENTER

Press ENTER to complete the IPL.

Rebuild the CP Nucleus

Note for NLS

If you would like to change the default system national language for CP you must rebuild and save the CP load map To do this, follow the steps in Appendix C, “Building the NLS Nuclei, Including UCENG” on page C-1 to change the PPF file if necessary. Then continue with “Rebuild the CP Nucleus” and “Verify the CP Load Map” on page 4-22. In these procedures substitute the national language $PPF file name for *ppfname*.

1 Make sure your machine environment is correct (XA), your storage is defined to at least 35M and the necessary disks are accessed. If your environment is not correct follow the instructions on “Step 1. Prepare the System for Service” on page 3-5 (steps 3 and 4).

2 Build the nucleus.

vmfbld ppf *ppfname* **cp cpload (all setup**

cpload is the CP load list.

Note

If you want the CP nucleus built as a punch text specify **punch** and the nucleus will be placed in your reader provided you have spooled your punch to your reader.

vmfbld ppf *ppfname* **cp cpload * nuctarg punch (all setup**

3 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Verify the CP Load Map

1 XEDIT the CPLOAD MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild the CP nucleus and save the map, see “Rebuild the CP Nucleus” on page 4-21 and Verify the CP Load Map. Then continue with “Rebuild the HELP Files in Uppercase (Optional)” on page 3-38.

xedit cpload map *fm-493*
====> set case mixed ignore

fm-493 is the file mode of the target disk which is where the CPLOAD MAP was placed when it was created.

====> all/unresol/
DMSXDC546E Target not found

This response means that no unresolved symbols were found.

```
====> all/undefin/
DMSXDC546E Target not found

====> qquit
```

This response means that no undefined symbols were found.

Note

You have now finished the detailed build steps for CP. Go to "Rebuild the HELP Files in Uppercase (Optional)" on page 3-38 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: GCS

Prepare to Build

Note

When entering the component name, or *compname* in this procedure, if you installed GCS in the Shared File System, use **gcssfs** as the *compname*, instead of **gcs**. Moving GCS from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.

The notation for the component names is **gcs|gcssfs**. Enter **gcs** or **gcssfs**. Do not enter the **|**.

Determine Which Build Lists Need to be Processed

1 To determine which build steps to perform:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-23, and then run the corresponding build steps in the order that they appear in the table.

Table 4-23. GCS Build Lists	
Build List	**Build Procedure**
GCTBLHLP GCTBLINF GCTBLRAS GCTBLRST	"Select and Copy Serviced Files" on page 4-24
GCTGPI	"Update the Macro Libraries" on page 4-24
GCTBLPRP	"Rebuild the Load Library" on page 4-25
GCTSBHLP	"Process the Segment Build List" on page 4-26
GCTLOAD	"Rebuild the GCS Nucleus" on page 4-26

Select and Copy Serviced Files

Table 4-24 contains build lists for GCS that require the select and copy build functions.

Table 4-24. Build Lists for GCS Select and Copy Functions		
Build List	**Description**	**Test/Production Disk**
GCTBLHLP	Build list for HELP files.	19D/19D
GCTBLINF	Build list for GCS/Dump Viewing Facility interface files.	493/193
GCTBLRAS	Build list for GCS Serviceability tools	493/193
GCTBLRST	Build list for miscellaneous GCS files	493/193

1 Enter the following command for each build list in Table 4-24 that was in the $VMFBLD $MSGLOG.

vmfbld ppf esa gcs|gcssfs *bldlist* **(serviced**

bldlist is the build list in Table 4-24.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Update the Macro Libraries

Note

If you already updated the MACLIBs as part of reworking local service earlier in this component step, skip this step, and go to "Rebuild the Load Library" on page 4-25.

Table 4-25 on page 4-25 lists the GCS MACLIBs that may need to be rebuilt.

Table 4-25. Build List for GCS MACLIBs

Build List	Description	Test/Production
GCTGPI	Build list for the GCTGPI MACLIB.	493/193

1 Enter the VMFBLD command to rebuild the GCTGPI MACLIB. When servicing a MACLIB using VMSES/E, ensure that your A-disk has enough free space to hold two copies of the MACLIB. This is to allow for the staging of the MACLIB, updating the serviced MACRO and COPY files and for the MACLIB compression.

```
vmfbld ppf esa gcs|gcssfs gctgpi (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Load Library

Table 4-26 contains the load library that may need to be rebuilt.

Table 4-26. Build List for GCS Load Library

Build List	Description	Test/Production Disk
GCTBLPRP	Build list for the PROPMX LOADLIB.	490/190

1 Enter the VMFBLD command to rebuild the load library for each build list that needs to be rebuilt.

```
vmfbld ppf esa gcs|gcssfs gctblprp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-27. Build List for GCS Segment Build List

Build List	Description
GCTSBHLP	Build list for GCS files that reside on the HELP segment

1 Enter the following command to process the HELP segment:

```
vmfbld ppf esa gcs|gcssfs gctsbhlp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the GCS Nucleus

Note

Parts of the REXX/VM component are contained in the GCS nucleus. You must receive and apply REXX/VM service (if any) before you rebuild the GCS nucleus.

Note for NLS

If you would like to change the default system national language for GCS you must rebuild and IPL the GCS nucleus to contain the language. To do this, follow the steps in Appendix C, “Building the NLS Nuclei, Including UCENG” on page C-1 to change the PPF file if necessary. Then continue with “Rebuild the GCS Nucleus” and “IPL the GCS Nucleus” on page 4-27. In these procedures substitute the national language $PPF file name for *ppfname*.

1 Make sure your machine environment is correct (XA), your storage is defined to at least 35M and the necessary disks are accessed. If your environment is not correct follow the instructions on “Step 1. Prepare the System for Service” on page 3-5 (steps 3 and 4).

2 Make any other changes that you want to include in the nucleus, such as changes to the GCS configuration file using the GROUP EXEC. For more information, see *VM/ESA: Installation Guide*.

3 Build the nucleus.

spool prt *
spool pun *

Direct the output of the virtual printer and punch to your virtual reader.

vmfbld ppf *ppfname* **gcs|gcssfs gctload (all setup**

gctload is the build list for the GCS nucleus. If you are not using the default, substitute your nucleus name for **gctload**.

```
SYSTEM LOAD DECK COMPLETE
PUN FILE spoolid TO MAINT COPY 001 NOHOLD
```

The punch file containing the nucleus is spooled to your reader. Record the spool file ID from this message, as it will be used to save the nucleus in the saved segment.

4 Check the results of the build. Use VMFVIEW to examine the $VMFBLD $MSGLOG file to verify that the correct levels of the text decks were loaded into the nucleus.

vmfview build

IPL the GCS Nucleus

1 Make the file that contains the new nucleus the first file in your reader and IPL the new nucleus.

Note: This will put GCS into production.

order rdr *spoolid*

spoolid is the spool ID of the nucleus file that you recorded earlier. In the previous example, the spool ID is 0025.

ipl 00c clear

Save the nucleus in your saved segment.

```
NAMED SAVED SYSTEM (NSS) GCS
WAS SUCCESSFULLY SAVED IN FILEID 0100
```

This message indicates that your nucleus was successfully saved.

```
RDR FILE spoolid SENT FROM MAINT
```

Record the spool file ID from this message. It identifies the file containing the nucleus load map that was spooled to your virtual reader.

ipl 190 clear
VM/ESA V2.4.0 *mm/dd/yy hh:mm*

Restore your CMS environment.

ENTER

Press ENTER to complete the IPL.

2 Access the necessary VMSES/E disks:

access *address* **b**
access 51d d

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

3 Save the load map on the test system tools disk (493).

access 493 *fm-493*

fm-493 is any file mode for the 493 disk.

receive *spoolid* **gcsnuc map** *fm-493* **(olddate replace**

spoolid is the spool file ID of the GCS load map that was sent to your reader when you built the GCS nucleus.

vmfcopy gcsnuc map *fm-493* **= = = (prodid 2vmvml40%gcs replace pack**

4 XEDIT the GCSNUC MAP and check for unresolved or undefined references. Unresolved or undefined symbols sometimes, but not always, indicate an error. Make sure you understand the reason for any unresolved or undefined symbols before going on. If you need to rebuild and IPL the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27. Then return to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-47.

```
xedit gcsnuc map
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found
```

This response means that no unresolved symbols were found.

```
====> all/undefin/
DMSXDC546E Target not found
```

This response means that no undefined symbols were found.

```
====> qquit
```

release *fm-493*

fm-493 is the file mode of the 493 disk.

Note

You have now finished the detailed build steps for GCS. Go to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-47 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: Dump Viewing Facility

Determine Which Build Lists Need to be Processed

1 To determine which build steps need to be performed:

vmfview build

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-28, and then run the corresponding build steps in the order that they appear in the table.

Table 4-28. Dump Viewing Facility Build Lists	
Build List	**Build Procedure**
HCSBL493 HCSBLHLP HCSBLMSG HCSBLSRC	“Select and Copy Serviced Files” on page 4-29
HCSMLOAD HCSSLOAD	“Rebuild the Executable MODULEs” on page 4-29
HCSSBHLP	“Process the Segment Build List” on page 4-30

Select and Copy Serviced Files

Table 4-29 contains build lists for Dump Viewing Facility that require the select and copy build functions.

Table 4-29. Build Lists for Dump Viewing Facility Select and Copy Functions		
Build List	**Description**	**Test/Production Disk**
HCSBL493	Build list for EXEC, XEDIT and other files for the 493/193 disks.	493/193
HCSBLHLP	Build list for HELP files.	19D/19D
HCSBLMSG	Build list for text files for the 493/193 disks	493/193
HCSBLSRC	Build list for source files.	493/193

1 Enter the following command for each build list in Table 4-29 that was in the $VMFBLD $MSGLOG.

vmfbld ppf esa dv *bldlist* **(serviced**

bldlist is the build list in Table 4-29.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Executable MODULEs

If any Dump Viewing Facility executable MODULEs have been serviced, they must be rebuilt.

Table 4-30. Build Lists for Dump Viewing Facility Executable MODULEs

Build List	Description	Test/Production Disk
HCSMLOAD	Build list for VMFBLD to build DVF MODULEs on the tools disk.	493/193
HCSSLOAD	Build list for VMFBLD to build DVF MODULEs on the system disk.	490/190

1 Enter the VMFBLD command to rebuild all of the MODULEs in the build list that contain text decks that have been serviced.

vmfbld ppf esa dv *bldlist* **(serviced**

bldlist is the build list in Table 4-30.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-31. Build List for Dump Viewing Facility Segment Build List

Build List	Description
HCSSBHLP	Build list for Dump Viewing Facility files that reside on the HELP segment

1 Enter the following command to process the HELP segment:

vmfbld ppf esa dv hcssbhlp (serviced

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Note

You have now finished the detailed build steps for Dump Viewing Facility. Go to "Rebuild the HELP Files in Uppercase (Optional)" on page 3-53 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: TSAF

Prepare to Build

Note

When entering the component name, or *compname* in this procedure, if you installed TSAF in the Shared File System, use **tsafsfs** as the *compname*, instead of **tsaf**. Moving TSAF from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.

The notation for the component names is **tsaf|tsafsfs**. Enter **tsaf** or **tsafsfs**. Do not enter the **|**.

Determine Which Build Lists Need to be Processed

1 To determine which build steps need to be performed:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-32, and then run the corresponding build steps in the order that they appear in the table.

Table 4-32. TSAF Build Lists

Build List	Build Procedure
ATSBLHLP ATSBLINF ATSBLMES ATSBLSRC	"Select and Copy Serviced Files" on page 4-32
ATSMLOAD	"Rebuild the Executable MODULEs" on page 4-32
ATSSBHLP	"Process the Segment Build List" on page 4-33

Select and Copy Serviced Files

Table 4-33 contains build lists for TSAF that require the select and copy build functions.

Table 4-33. Build Lists for TSAF Select and Copy Functions

Build List	Description	Test/Production Disk
ATSBLHLP	Build list for HELP files.	19D/19D
ATSBLINF	Build list for TSAF/Dump Viewing Facility interface files.	493/193
ATSBLMES	Build list for message text files	493/193
ATSBLSRC	Build list for new/replacement source files.	493/193

1 Enter the following command for each build list in Table 4-33 that was in the $VMFBLD $MSGLOG.

vmfbld ppf esa tsaf|tsafsfs *bldlist* **(serviced** *bldlist* is the build list in Table 4-33.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. Form information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Rebuild the Executable MODULEs

Table 4-34 describes the build list used to rebuild executable MODULEs.

Table 4-34. Build Lists for TSAF Executable MODULEs

Build List	Description	Test/Production Disk
ATSMLOAD	Build list for the executable TSAF MODULEs	490/190

1 Enter the VMFBLD command to rebuild all of the MODULEs in the build list that contain text decks that have been serviced.

vmfbld ppf esa tsaf|tsafsfs atsmload (serviced

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview build

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-35. Build List for TSAF Segment Build List

Build List	Description
ATSSBHLP	Build list for TSAF files that reside on the HELP segment

1 Enter the following command to process the HELP segment:

```
vmfbld ppf esa tsaf|tsafsfs atssbhlp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Note

You have now finished the detailed build steps for TSAF. Go to "Rebuild the HELP Files in Uppercase (Optional)" on page 3-60 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: AVS

Prepare to Build

Note

When entering the component name, or *compname* in this procedure, if you installed AVS in the Shared File System, use **avssfs** as the *compname*, instead of **avs**. Moving AVS from a CMS minidisk to an SFS directory was decided during the VM/ESA installation process.

When issuing the VMFBLD command, use the *compname* **avs** or **avssfs** if you have a GCS/XA environment.

The notation for the component name is **avs|avssfs**. Enter **avs** or **avssfs**. Do not enter the **|**.

Determine Which Build Lists Need to be Processed

1 To determine which build steps need to be performed:

```
vmfview build
```

2 Find the names of the build lists that contain objects that need to be built, as listed in Table 4-36, and then run the corresponding build steps in the order that they appear in the table.

Table 4-36. AVS Build Lists

Build List	Build Procedure
AGWBLHLP AGWBLINF AGWBLSRC	"Select and Copy Serviced Files"
AGWBLLIB AGWBLUTL	"Rebuild the Load Libraries" on page 4-35
AGWSBHLP	"Process the Segment Build List" on page 4-35

Select and Copy Serviced Files

Table 4-37 contains build lists for AVS that require the select and copy build functions.

Table 4-37. Build Lists for AVS Select and Copy Functions

Build List	Description	Test/Production Disk
AGWBLHLP	Build list for HELP files.	19D/19D
AGWBLINF	Build list for AVS/Dump Viewing Facility interface files.	493/193
AGWBLSRC	Build list for new/replacement source files.	493/193

1 Enter the following command for each build list in Table 4-37 that was in the $VMFBLD $MSGLOG.

```
vmfbld ppf esa avs|avssfs bldlist (serviced
```

bldlist is the build list in Table 4-37.

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Rebuild the Load Libraries

Table 4-38 lists the AVS load libraries that may need to be rebuilt.

Table 4-38. Build Lists for AVS Load Libraries

Build List	Description	Test/Production Disk
AGWBLLIB	Build list for AGW	493/193
AGWBLUTL	Build list for AGWUTIL	493/193

1 To update the AGWUTIL LOADLIB, enter the following VMFBLD command:

```
vmfbld ppf esa avs|avssfs agwblutl (serviced
```

2 To update the AGW LOADLIB, enter the following VMFBLD command:

```
vmfbld ppf esa avs|avssfs agwbllib (serviced
```

3 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed during the task Step 11. Place Your Serviced Components into Production.

Table 4-39. Build List for AVS Segment Build List

Build List	Description
AGWSBHLP	Build list for AVS files that reside on the HELP segment

1 Enter the following command to process the HELP segment:

```
vmfbld ppf esa avs|avssfs agwsbhlp (serviced
```

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

```
vmfview build
```

Note

You have now finished the detailed build steps for AVS. Go to “Rebuild the HELP Files in Uppercase (Optional)” on page 3-67 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Chapter 5. Local Service and Modifications — Overview

This chapter contains an overview of the step-by-step procedures for applying local service to VM/ESA components. Local service and local modifications are defined as any service that is applied to your VM/ESA system that was **not** supplied by IBM on a COR or RSU tape. For the instructions, see Chapter 6, “Procedures for Local Service and Modifications” on page 6-1.

This chapter also contains an overview of the step-by-step procedures for reworking local service. Existing local service can be affected by new IBM service which cause the need to rework the local service. For the instructions, see Chapter 7, “Reworking Local Service and Modifications” on page 7-1.

Examples for applying local modifications to the CP load list and CMSINST can be found in the appendix. If you want to update the CP load list see Appendix F, “Updating the CP Load List” on page F-1 for instructions on using the GENCPBLS command. If you plan to change which EXECs are placed in the CMS installation segment see Appendix E, “Local Modification Example for CMSINST” on page E-1.

Introduction

Local service and **local modifications** are defined as any **service** or **software change** that is applied to your VM/ESA system that was not supplied by IBM on a COR or RSU tape.

Attention

The application of local service can be a complicated and error-prone procedure because of the many variables that are involved. IBM strongly advises its customers to order service on a COR or RSU tape through the IBM Support Center whenever possible.

When it is absolutely necessary to apply service from IBM before it is available on a COR tape, or when you need a local modification to tailor your system environment, you must apply the service locally. This includes updates supplied to you by other vendors.

The local Version Vector Table must be updated in order for VMSES/E to process your local modifications.

Local service can be placed in two categories: IBM local service and customer local modifications.

IBM local service includes any service that you receive from IBM that is not on a COR or RSU tape.

When a severe problem arises and you cannot wait for a COR or RSU tape, you can get emergency service from the change team. This service can be sent to you on a tape or read to you over the phone. In some cases, a member of the change team can place the fix in the library that is accessed by the APARFIX command on ServiceLink. In any case, the service is **not** in the format required by the VMFREC

and VMFAPPLY EXECs. To receive and apply this service, you **must** do it manually, using the instructions in this chapter.

Customer local modifications include any software changes that are **not** supplied by VM/ESA. Customer local modifications tailor your VM/ESA system. These updates can be supplied by other IBM licensed products or by other vendor products. For example, if you have RACF®/VM you will have a customer local modification because RACF/VM has a “mod” to CP.

There are three ways that parts can be serviced. Source-maintained parts are serviced by changing the source, with an update file, an AUX file, and a CNTRL file. Replacement-maintained parts are serviced by replacing the part with an updated version. Thirdly, when necessary, some parts (such as text files) can be serviced by directly changing the object code they contain. Local service for source-maintained and replacement-maintained parts is described in this chapter. The procedure for changing object code is described in Appendix H, “Apply Changes Directly to Object Code” on page H-1.

Rework local service may be necessary if the service you receive from IBM affects the existing local service. You will be notified of this possibility when you install a COR or RSU tape. If this happens, re-evaluate the local service and then, if needed, rework and rebuild the affected parts.

Overview for Local Service Procedure

The following is an overview of the steps in the local service procedure. For the instructions see Chapter 6, “Procedures for Local Service and Modifications” on page 6-1.

For each modified part in a component:

1. Prepare for Local Service or Modifications.

   Access the component's service disks and the VMSES/E BUILD disk (5E5 by default). If necessary, modify the CNTRL file.

2. Receive Local Service or Modifications.

   If you have IBM local service, load the service to the LOCALMOD disk for your product.

3. Apply Local Service to Source-Maintained parts.

   a. Add an Update Record to the AUX file.

   b. Create an Update File if it is not shipped.

   c. Special Processing for MACROs with ASSEMBLE files.

      1) Update the local version vector table with the VMFSIM CHKLVL command and the LOGMOD option.

      2) Determine what MACLIBs need to be rebuilt.

      3) For each ASSEMBLE file which uses the updated macro and does not have another change in this local service, create a dummy update for the ASSEMBLE file.

4. Apply Local Service to Replacement-Maintained parts.

   a. Create or Copy the Replacement Part.

b. Special Processing for MACROs with ASSEMBLE files.

1) Determine what MACLIBs need to be rebuilt.

2) For each ASSEMBLE file which uses the updated macro and does not have another change in this local service, create a dummy update for the ASSEMBLE file.

Repeat 3 - 4

For each modified part in this component, repeat step 3 or 4.

5. Rebuild Objects

a. Rebuild Source files.

b. Create compiled REXX parts.

c. Create a Replacement Part from $Source files. Create a replacement part with the VMFEXUPD command. This command will place the output on the LOCALMOD disk, update the local Version Vector Table (VVT) and add an entry to the $SELECT file.

d. Rebuild MACLIBs with the VMFBLD command.

e. Create a replacement part from updated ASSEMBLE files. Rebuild ASSEMBLE files with the VMFHLASM command. This command will place the output on the LOCALMOD disk, update the local Version Vector Table (VVT) and add an entry to the $SELECT file.

f. Create replacement part from replaced ASSEMBLE files.

g. Create a replacement part from modified National Language files with the VMFNLS command. This command will place the output on the LOCALMOD disk, update the local Version Vector Table (VVT) and add an entry to the $SELECT file.

h. Rebuild any remaining objects.

Overview for Rework Local Service Procedure

The following is an overview of the steps for the rework local service procedure. For the instructions see Chapter 7, “Reworking Local Service and Modifications” on page 7-1.

1. Prepare to rework local service or modifications.
2. Rework Local Service to Source-Maintained parts.
3. Rework Local Service to Replacement-Maintained parts.
4. Rebuild the Objects.

Definitions for Variables

During the procedures to apply local service or rework local service the following common abbreviations will be used:

aparnumber — *aparnumber* is the APAR number.

appid	*appid* specifies the file name of the version vector table and the $SELECT file. This is the first value specified on the :APPID. tag in the PPF for the component you are servicing. See Table 5-1 on page 5-6.
auxft	*auxft* is the file type of the local AUX file. AUXLIXX is used for REXX files. AUXLGCT is used for GCS files. The default for the other components is AUXLCL. The defaults will be changed depending on what level is used in the CNTRL file. See Table 5-1 on page 5-6.
cftabbrev	*cftabbrev* is the abbreviation for the compiled file type: **CEX** for EXEC files or **CAE** for XEDIT files.
comment	*comment* is any comment that you want to add when you XEDIT a file.
compname	*compname* is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, CMS, CP, GCS, DV, TSAF, and AVS.
ctlfile	*ctlfile* is the file name of the control file. This is specified on the :CNTRL. tag in the PPF for the component you are servicing. See Table 5-1 on page 5-6.
fm-local	*fm-local* is the file mode of the LOCALMOD disk for the component you are servicing.
fn	*fn* is the file name of the part.
ft	*ft* is the file type of the part.
ftabbrev	*ftabbrev* is the 3 character abbreviation for the file type of the part that was modified. To obtain a *ftabbrev* see “Obtaining File Type Abbreviations” on page 5-6.
modid	*modid* is the local modification number. It should begin with L, and is followed by up to 4 alphanumeric characters that identify the local modifications. For example, LFIX1 is valid. This must be in uppercase. The *modid* is also used to create the file type of the object generated when the local modification is compiled.

curmodid	*curmodid* is the current local modification number.
ftabbrev-modid	*ftabbrev-modid* is the file type of the compiled part and consists of the *ftabbrev* concatenated with the *modid*. The *ftabbrev* is the 3 character file type abbreviation. See the *modid* abbreviation for more information.
ftabbrev-ptfnumber	*ftabbrev-ptfnumber* is the file type of the IBM replacement service to a part and consists of the *ftabbrev* concatenated with the *ptfnumber*. The *ftabbrev* is the 3 character file type abbreviation. The *ptfnumber* is the PTF number.
lc*modid*	**lc***modid* is **lc** concatenated with *modid* and is the local modification identification. The first 2 characters should be LC, followed by the *modid*. For example, LCLFIX1 is valid. This must be in uppercase.
ppfname	*ppfname* is the file name of the PPF being used. *ppfname* is shown as **esa**. If you have your own *ppfname*, substitute that name for the **esa** name shown in the commands.
ptfnumber	*ptfnumber* is the PTF number.
realft	*realft* is the real file type of the part. For example, MACRO or COPY.
sftabbrev	*sftabbrev* is the abbreviation for the uncompiled file type of a compiled part: **SXE** for EXEC files or **SXD** for XEDIT files.
svclvl	*svclvl* is a service level indicator. This must be in uppercase. This is the second token in the AUX file entries and is usually LCL.
updateft	*updateft* is the file type of the update file containing your local modifications to source files. This must be in uppercase. *updateft* is not used for replacement parts. You will need to define this if the update file was not supplied by IBM software support.

vvtft — *vvtft* is the file type of the local VVT file. VVTLIXX is used for REXX files. VVTLGCT is used for GCS files. The default for the other components is VVTLCL. The defaults will be changed depending on what level is used in the CNTRL file. See Table 5-1 on page 5-6.

Obtaining File Type Abbreviations

To obtain a file type abbreviation, enter the following VMFSIM command:

vmfsim query vm sysabrvt tdata :realft *ft*

The abbreviation is returned on the :ABBRFT tag. If more than one abbreviation is returned, you must use the one that is used by the part that you are modifying. To determine if an abbreviation is the correct one for a particular part, enter the following command:

vmfqobj *ppfname compname* **tdata :part** *fn ftabbrev*

Default Control, Apply and AUX File Identifiers

Use the control file name, default apply identifier, and local AUX and VVT file type identifiers listed in Table 5-1 throughout the procedure unless you have changed them.

Table 5-1. Default Control, Apply, and AUX File Identifiers

Component	*ctlfile*	*appid*	*auxft*	*vvtft*
VMSES/E	VMFVM	2VMVMK40	AUXLCL	VVTLCL
REXX/VM	IXXVM	2VMVMF40	AUXLIXX	VVTLIXX
CMS	DMSVM	2VMVMA40	AUXLCL	VVTLCL
CP	HCPVM	2VMVMB40	AUXLCL	VVTLCL
GCS	GCTVM	2VMVML40	AUXLGCT	VVTLGCT
Dump Viewing Facility	HCSVM	2VMVMI40	AUXLCL	VVTLCL
TSAF	ATSVM	2VMVMH40	AUXLCL	VVTLCL
AVS	AGWVM	2VMVMD40	AUXLCL	VVTLCL

Note: When assembling CMS Pipelines parts (any part with a file name of FPL*xxx* plus the following parts: DMSCRX, DMSPFP, DMSPPP, DMSPQI, DMSPRN, DMSXFB, MDATPEEK, and RITA), use the FPLVM control file instead of the default CMS control file, DMSVM.

Chapter 6. Procedures for Local Service and Modifications

Use these procedures while applying local service or local modifications. Use the procedures in Chapter 7, “Reworking Local Service and Modifications” on page 7-1 when reworking existing local service or local modifications.

This chapter refers to “Definitions for Variables” on page 5-3 for the definitions of variables used in this chapter. It is recommended that you review Chapter 5, “Local Service and Modifications — Overview” on page 5-1 if you are unfamiliar with local service and local modifications.

Step 1. Prepare for Local Service or Modifications

1 As a precaution, do a complete backup of your system. See the *VM/ESA: Installation Guide* for instructions on storing a backup copy of the VM/ESA system to tape.

2 Set up the correct virtual machine environment and IPL.

set machine xa
System reset.
System = XA

If your machine environment is already in the correct mode, you will not see all of these displayed messages.

define storage *nn***M**
STORAGE CLEARED - SYSTEM RESET
STORAGE = nnM

Define the storage to at least 35MB. If you will be servicing GCS, make sure to define your storage to be at least as large as the value you defined for HIGH COMMON END storage (default 19MB) during the installation procedures. For more information, see *VM/ESA: Installation Guide*.

Defining storage causes a system reset.

ipl 190 clear

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

ENTER

Press ENTER to complete the IPL.

3 If you are creating local service to source-maintained parts and the source code was not loaded during the installation of VM/ESA, you will need to load the code now. See the *VM/ESA: Installation Guide* for instructions.

4 Access the necessary VMSES/E disks:

access *address* **b**
access 51d d

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

5 If your PPF file has changed, enter the following command:

vmfppf esa *compname*

esa is the PPF file name. If you have an override to **esa**, use it instead.

compname is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, CMS, CP, GCS, DV, TSAF, and AVS.

6 Set up your access order by entering the following command. Using the command output, make a note of the file modes of the LOCALMOD disk, the alternate APPLY disk, and the BASE disks for this component. The LOCALMOD disks and alternate APPLY disk must be accessed R/W.

vmfsetup esa *compname*

7 Select a *modid* for this local modification. It will be used for all parts affected by this change. The *modid* must be unique. See “Definitions for Variables” on page 5-3 for the definition of a *modid*.

8 If your control file (identified in the PPF on the :CNTRL. tag) does not contain an AUX record for a local AUX file (see Table 5-1 on page 5-6 for the default file types) you will need to add one. Look at DMSVM CNTRL for an example. Use the local modification procedure “Step 4. Apply Local Service to Replacement-Maintained Parts” on page 6-7 to modify your control file and identify the local modification in the local version vector table (VVT). Then rebuild the control file before continuing with the local modification.

9 Rebuild source files.

If you have applied IBM supplied service and are applying local service, you need to insure that the source files are updated (in case the files were serviced by IBM). GCS does not ship source files so this step is not necessary for this component. For all other components you are processing, issue the following command.

vmfbld ppf esa *compname bldlist* **(serviced**

compame and *bldlist* are the names of the component and build list which can be:

VMSES	VMFBLSRC
REXX	IXXBLSRC
CMS	DMSBLSRC
CP	HCPBLSRC
DV	HCSBLSRC
TSAF	ATSBLSRC
AVS	AGWBLSRC

Step 2. Receive Local Service or Modifications

To receive local service that has been supplied to you:

1 If the service is emergency service supplied by telephone or as hard copy, your IBM software support representative will give you the information you need to create the emergency fix. Make sure that you copy the files to the LOCALMOD disk.

2 If the service is supplied on a magnetic tape, load it onto the LOCALMOD disk (*fm-local*).

a Attach a tape drive to your virtual machine as 181.

attach *rdev* *** 181**

rdev is the real device number of the tape drive.

b Mount and ready the tape.

c Load the service to the LOCALMOD disk.

vmfplc2 load * * *fm-local* **(eot disk**

The tape file is in VMFPLC2 format. The VMFPLC2 command creates a file called TAPE MAP A. The TAPE MAP file lists the file names and file types of all of the files that were loaded to the LOCALMOD disk.

Step 3. Apply Local Service to Source-Maintained Parts

Note

Complete Step 3 only if you are applying a change to a source-maintained part using update files. If you are applying a change to a replacement-maintained part, skip to "Step 4. Apply Local Service to Replacement-Maintained Parts" on page 6-7.

Making Source Updates

To make a source update, you must:

- Add an update record to the LOCAL AUX file.
- Generate a source update file that contains your updates. (If you are making a circumventive change, you might obtain this file from IBM.)
- Generate the new level of the part. The new level is generated when objects are rebuilt during "Step 5. Rebuild Objects" on page 6-9.

> **Note**
>
> If you are applying a change to the HCPMDLAT MACRO, the instructions for updating the CP Load List are in Appendix F, "Updating the CP Load List" on page F-1.

Guidelines for Using Update Files

Here are some guidelines for using update files for source updates and patches:

1. Keep each fix in a separate update file. You can have several update files containing one fix, but you must never have several fixes in one update file.

   Each fix should have a unique identifier for control purposes. This identifier is the file type of the update file. If the same fix applies to several parts, there should be a update file for each part, all with the same file type.

2. Keep all local fix descriptions for the same part in the same AUX file, unless a fix applies to a different control file level.

   You should not distribute local fixes for the same part over AUX files (different control file levels) arbitrarily. Local service can be distributed over several control files to maintain different service levels with a single AUX and update file structure. You can build each level from a different control file that contains only the desired level identifiers.

3. Never place local fixes in AUX files from IBM. In other words, keep your local service separate from IBM service. Local service should be easily distinguished from IBM service and should always be applied last.

4. Patches to text files should be applied only when no source file is available. If you do apply text file patches when source code is available, you should convert them to source updates and reassemble them before you move them from a test environment to your production system.

   Building local source updates on top of local text file patches for the same part will lead to confusion.

5. Do not place the names of update files in the main control file. Place update file names in AUX files.

Step 3a. Add an Update Record to the AUX File

1. Update or create the AUX file (*fn auxft*, where *fn* is the name of the source-maintained part being updated and *auxft* is the file type of the local AUX file) and add an entry for the update file.

   Refer to "Definitions for Variables" on page 5-3 to help you when creating the file names.

```
xedit fn auxft fm-local (noprof
====> input updateft svclvl lcmodid  comment
====> file
```

Step 3b. Create or Copy the Update File

Note

You must complete either substep1 or substep 2. If you have the update file use substep 2. If you need to create the update file use substep 1. In either case the new level of the part will be generated in “Step 5. Rebuild Objects” on page 6-9.

1 If you need to create the update, use these commands.

a XEDIT the source file with the CTL option.

xedit *fn ft fm* **(ctl** *ctlfile*

fn, *ft* and *fm* are the file name, file type and file mode of the source file you are modifying.

b Make your changes to the displayed source file. The original source file is **not** changed.

c When you complete your changes, save them on the LOCALMOD disk. When you enter the FILE command in XEDIT, all of your changes are placed in the update file, *fn updateft*.

====> file = = *fm-local*

d Continue on to “Step 3c. Special Processing for Macros with ASSEMBLE Files.”

2 If you have the update, copy it to the local modification disk.

copyfile *fn ft fm* **= =** *fm-local*

fn ft fm are the file name, file type, and file mode of your update file.

Step 3c. Special Processing for Macros with ASSEMBLE Files

Note

You must complete this step only if you are installing local service to a source-maintained macro.

1 Update the Local Version Vector Table. Enter the following command to identify the local modification in the local version vector table for the component. This updates the local version vector table with the AUX entries for the part.

vmfsim chklvl esa *compname* **tdata :part** *fn realft* **(logmod**

realft is the real file type (for example, MACRO or COPY). The *realft* is used to indicate to the VMFBLD command that this is a source-maintained macro.

2 Determine what MACLIBs need to be rebuilt.

vmfqobj esa *compname* **tdata :object** *fn* **:libname**

:OBJECT *bldlist.fn*
:LIBNAME *libname*

fn is the name of the macro.

bldlist is the name of a build list of a MACLIB that needs to be rebuilt. Record this build list name for use in "Step 5d. Rebuild MACLIBs" on page 6-11

3 Update the $SELECT file for the macro.

vmfrepl *fn ft* **esa** *compname* **($select nocopy filetype** *ft*

ft is the file type of the macro (for example, MACRO or COPY).

4 If you have ASSEMBLE files that use the preceding macro and the ASSEMBLE files do not already have an update in this local modification, then you must identify the ASSEMBLE files as part of this local modification. This is done by creating a dummy update (consisting of an update comment) for each ASSEMBLE file that uses the updated MACRO.

a Update or create the AUX file (*fn auxft*, where *fn* is the name of the source-maintained part being updated and *auxft* is the file type of the local AUX file) and add an entry for the update file.

Refer to "Definitions for Variables" on page 5-3 to help you when creating the file names.

xedit *fn auxft fm-local* **(noprof**
====> input *updateft svclvl* **lc***modid comment*
====> file

fn is the file name of the ASSEMBLE file.

b Create the dummy update file, *fn updateft*, on the LOCALMOD disk.

xedit *fn updateft fm-local* **(noprof**
====> input ./ * lc*modid comment*
====> file

fn is the file name of the ASSEMBLE file.

Step 4. Apply Local Service to Replacement-Maintained Parts

Note

Complete Step 4 only if you are applying a change to a replacement-maintained part. If you are applying a change to a source-maintained part, you should have completed "Step 3. Apply Local Service to Source-Maintained Parts" on page 6-3 and should now skip to "Step 5. Rebuild Objects" on page 6-9.

Step 4a. Create or Copy the Replacement Part

Note

You must complete either substep 1 or substep 2. If you need to create the modified part, use substep 1. If you have the modified part, use substep 2.

1 If you need to create the part, follow these steps.

a Copy the highest level of the part to the LOCALMOD disk, with the correct file type for the replacement part.

vmfrepl *fn ft* **esa** *compname* **($select logmod** *modid* **outmode localmod**

fn and *ft* are the real file name and file type of the part.

modid is the new local modification identifier for the part.

localmod is the symbolic name for the local disk in the :MDA section of the ESA PPF.

b XEDIT the part copied to the LOCALMOD disk.

xedit *fn ftabbrev-modid fm-local*

The file ID (*fn ftabbrev-modid fm-local*) was returned in message VMFREP2507I.

c Make your changes to the displayed file.

d When you complete your changes, save the replacement part.

====> file

e Continue on to "Step 4b. Special Processing for Macros with ASSEMBLE files" on page 6-8.

2 If you have the replacement part, copy the file to the local modification disk.

vmfrepl *fn ft* **esa** *compname rfn rft rfm* **($select logmod** *modid* **outmode localmod**

fn and *ft* are the real file name and file type of the part.

rfn, *rft* and *rfm* are the file name file type and file mode of your replacement part.

modid is the new local modification identifier for the part.

localmod is the symbolic name for the local disk in the :MDA section of the ESA PPF.

Step 4b. Special Processing for Macros with ASSEMBLE files

Note

You must complete this step only if you are installing local service to a replacement-maintained macro.

1 Use VMFQOBJ to determine what MACLIBs need to be rebuilt.

vmfqobj esa *compname* **tdata :object** *fn* **:libname**

:OBJECT *bldlist.fn*
:LIBNAME *libname*

fn is the name of the macro.

bldlist is the name of a build list of a MACLIB that needs to be rebuilt. Record this build list name for use in “Step 5d. Rebuild MACLIBs” on page 6-11.

2 If you have ASSEMBLE files that use the preceding macro and the ASSEMBLE files do not already have an update in this local modification, then you must identify the ASSEMBLE files as part of this local modification. This is done by creating a dummy update (consisting of an update comment) for each ASSEMBLE file that uses the updated MACRO.

a Update or create the AUX file (*fn auxft*, where *fn* is the name of the source-maintained part being updated and *auxft* is the file type of the local AUX file) and add an entry for the update file.

Refer to “Definitions for Variables” on page 5-3 to help you when creating the file names.

xedit *fn auxft fm-local* **(noprof**
====> input *updateft svclvl* **lc***modid comment*
====> file

fn is the file name of the ASSEMBLE file.

b Create the dummy update file, *fn updateft*, on the LOCALMOD disk.

xedit *fn updateft fm-local* **(noprof**
====> input ./ * lc*modid comment*
====> file

fn is the file name of the ASSEMBLE file.

Repeat 3 or 4

If you have additional parts to modify in this component, repeat “Step 3. Apply Local Service to Source-Maintained Parts” on page 6-3 or “Step 4. Apply Local Service to Replacement-Maintained Parts” on page 6-7. Then continue with “Step 5. Rebuild Objects” on page 6-9.

Step 5. Rebuild Objects

Note

When you have completed the preceding steps for all the local modifications for this component, then rebuild the objects.

Step 5a. Rebuild Source Files

Note

Complete this step only if you have created a replacement part for a source file in “Step 4. Apply Local Service to Replacement-Maintained Parts” on page 6-7.

To rebuild the source for the component, issue the following command. (GCS does not ship source files so this step is not necessary for this component.)

vmfbld ppf esa *compname bldlist* **(serviced**

compame and *bldlist* are the names of the component and build list which can be:

VMSES	VMFBLSRC
REXX	IXXBLSRC
CMS	DMSBLSRC
CP	HCPBLSRC
DV	HCSBLSRC
TSAF	ATSBLSRC
AVS	AGWBLSRC

Step 5b. Create Replacement Part from Modified $Source, Compiled REXX

Note

Complete this step only if you have created an update to a $source file for a compiled REXX part in “Step 3. Apply Local Service to Source-Maintained Parts” on page 6-3. You must complete substep 1 and either substep 2 or substep 3.

1 Create the uncompiled file.

vmfexupd *fn ft* **esa** *compname* **(outmode localmod $select logmod ftabbr** *sftabbrev*

fn ft is the file name and file type (without the leading $ from the file type).

sftabbrev is **SXE** for EXEC files or **SXD** for XEDIT files.

This creates the updated file named *fn sftabbrev-modid* on *fm-local*, updates the $SELECT file and VVT.

2 Create the compiled file using the REXX compiler. If you do not have the REXX compiler, go to substep 3.

a Use the REXXC command to create the compiled file.

rexxc *fn sftabbrev-modid fm-local* **(cexec(=** *cftabbrev-modid***)**

sftabbrev-modid is the file type of the output file from the previous step. It was identified in message VMFEXU2507I.

cftabbrev is the abbreviation for the compiled file type: **CEX** for EXEC files or **CAE** for XEDIT files.

b Use the VMFREPL command to update the $SELECT file and the local version vector table for the part. Skip substep 3.

vmfrepl *fn ft* **esa** *compname* **($select ftabbr** *cftabbrev* **logmod** *modid* **nocopy**

3 Create a replacement for the compiled file using the VMFREPL command if you do not have the REXX compiler. This command will also update the $SELECT file and the local version vector table.

vmfrepl *fn ft* **esa** *compname* **=** *sftabbrev-modid fmlocal* **($select ftabbr** *cftabbrev* **logmod** *modid* **outmode localmod**

sftabbrev-modid is the file type of the output file from the previous step. It was identified in message VMFEXU2507I.

cftabbrev is the abbreviation for the compiled file type: **CEX** for EXEC files or **CAE** for XEDIT files.

Step 5c. Create Replacement Part from Modified $Source, Not Compiled REXX

Note

Complete this step only if you have created an update to a $source file **other than** a compiled REXX part in “Step 3. Apply Local Service to Source-Maintained Parts” on page 6-3.

1 Update the file with the VMFEXUPD command.

vmfexupd *fn ft* **esa** *compname* **(outmode localmod $select logmod**

fn ft is the file name and file type (without the leading $ from the file type).

This creates the updated file named *fn ftabbrev-modid* on *fm-local*, updates the $SELECT file and the VVT.

Notes:

a. The file type of the source file must begin with a dollar sign ($), but **do not enter the $ in the command**.

b. The VMFEXUPD command uses the options HISTORY and SID.

c. If the file to be updated is an EXEC or XEDIT macro that is included in the CMS installation saved segment (default name CMSINST), use the NOCOMMENTS option to remove all comments and leading blanks. Then, rebuild the CMS installation saved segment to include the updated file.

d. If you are using RECEIVEX $XEDIT, rename the updated RECEIVEX XEDIT output to RECEIVE XEDIT.

e. The file $DASD$ $CONSTS cannot use the general command format; therefore, use the the following command with the additional option NOHIST:

vmfexupd $dasd$ consts esa cp (nohist outmode localmod $select logmod

fn ft is the file name and file type (without the leading $ from the file type).

If for some reason the VMFEXUPD command is unsuccessful, return to “Step 3b. Create or Copy the Update File” on page 6-5 to rework the update file.

Step 5d. Rebuild MACLIBs

Note

Complete this step if you have created a local modification (update or replacement) to a macro or to a MACLIB build list in “Step 3. Apply Local Service to Source-Maintained Parts” on page 6-3 or “Step 4. Apply Local Service to Replacement-Maintained Parts” on page 6-7 or if you have IBM supplied service to a MACLIB.

For each MACLIB that needs to be rebuilt enter the following command.

Note: You only need to rebuild a MACLIB once even if it contains multiple MACROs that were changed.

vmfbld ppf esa *compname bldlist* **(serviced**

bldlist is the name of a build list of a MACLIB that needs to be rebuilt.

Step 5e. Create a Replacement Part from Updated ASSEMBLE Files

Note

Complete this step only if you have created an update to an ASSEMBLE file in "Step 3. Apply Local Service to Source-Maintained Parts" on page 6-3 or if you created a dummy update for an ASSEMBLE file in "Step 3. Apply Local Service to Source-Maintained Parts" on page 6-3 or "Step 4. Apply Local Service to Replacement-Maintained Parts" on page 6-7.

For local modifications to the REXX/VM component, follow procedure 2; **for all other components,** follow procedure 1.

1 For components besides REXX/VM use the following command for ASSEMBLE files, except for the DMSTRT*cc* ASSEMBLE file. For the DMSTRT*cc* ASSEMBLE file, (where *cc* is a country code suffix), see "Step 5g. Create a Replacement Part from Modified National Language Files" on page 6-13. The command also updates the local version vector table with the AUX entries for the part, updates the $SELECT file and places the output file on the LOCALMOD disk.

If you are assembling a CMS Pipelines part (any part with a file name of FPL*xxx* plus the following parts: DMSCRX, DMSPFP, DMSPPP, DMSPQI, DMSPRN, DMSXFB, MDATPEEK, and RITA), use the second form of the VMFHLASM command which includes the CNTRL option.

vmfhlasm *fn* **esa** *compname* **(logmod $select outmode localmod**

OR

vmfhlasm *fn* **esa cms (logmod $select outmode localmod cntrl fplvm**

Continue with "Step 5f. Create a Replacement Part from Replaced ASSEMBLE Files" on page 6-13.

2 REXX/VM text decks are included in the CMS nucleus. Because of this, you cannot use the component name of REXX. Use instead the component name of CMS with the VMFHLASM command to ensure the correct object naming.

a You need to LOGMOD the local update in the REXX version vector table (VVT) by issuing the following command. This will allow VMFAPPLY to notify you when IBM service is shipped for the part that is modified.

vmfsim chklvl esa rexx tdata :part *fn* **txc (logmod**

b Issue the VMFHLASM command with CMS to ASSEMBLE the file for inclusion in the CMS nucleus. This also updates the $SELECT file and places the output file on the LOCALMOD disk.

```
vmfhlasm fn esa cms ($select outmode localmod
```

c In "Step 5h. Rebuild Remaining Objects" on page 6-14 you must rebuild CMS, but it is not necessary to rebuild REXX unless you have created a local modification to a REXX part other than an ASSEMBLE file.

Step 5f. Create a Replacement Part from Replaced ASSEMBLE Files

Note

Complete this step only if you have created a replacement ASSEMBLE file in "Step 4. Apply Local Service to Replacement-Maintained Parts" on page 6-7.

1 Enter the following command to update the local version vector table for the assembled text file:

```
vmfrepl fn text esa compname (logmod modid nocopy
```

modid is the same local modification identifier that you used for the ASSEMBLE part in "Step 4. Apply Local Service to Replacement-Maintained Parts" on page 6-7.

2 Enter the VMFHLASM command to assemble the file. The command places the output file on the LOCALMOD disk and updates the $SELECT file.

```
vmfhlasm fn esa compname (nockgen $select outmode localmod
```

Step 5g. Create a Replacement Part from Modified National Language Files

Note

Complete this step only if you have local modifications to any of the following national language files:

- *fncc* REPOS file
- *fncc* DLCS file
- DMSTRT*cc* ASSEMBLE file (where *cc* is a country code suffix).

1 Use the VMFNLS EXEC to generate the text deck for each national language file. This also updates the local version vector table with the AUX

entries for the part, updates the $SELECT file and places the output file on the LOCALMOD disk.

vmfnls *fn ft* **esa** *compname* **(logmod $select outmode localmod**

fn and *ft* are the file name and file type of the source file.

For more information about the VMFNLS EXEC, see *VM/ESA: VMSES/E Introduction and Reference*.

Step 5h. Rebuild Remaining Objects

You need to continue by building the rest of the objects affected by the local modification. To do this, go to the build section for the component you are processing and complete that section. Use the following table to find the appropriate build section.

Table 6-1. Build Section Reference

Component	Build Section
VMSES/E	“Prepare to Build” on page 3-10
REXX/VM	“Prepare to Build” on page 3-17
CMS	“Prepare to Build” on page 3-24
CP	“Prepare to Build” on page 3-34
GCS	“Prepare to Build” on page 3-42
Dump Viewing Facility	“Prepare to Build” on page 3-51
TSAF	“Prepare to Build” on page 3-58
AVS	“Prepare to Build” on page 3-65

Chapter 7. Reworking Local Service and Modifications

Use these procedures while reworking local service or local modifications. Reworking local service may be necessary if the service you receive from IBM affects the existing local service. You will be notified of this possibility when you install a COR or RSU tape. If this happens, re-evaluate the local service and then, if needed, rework and rebuild the affected parts.

This chapter refers to “Definitions for Variables” on page 5-3 for the definitions of variables used in this chapter. It is recommended that you review Chapter 5, “Local Service and Modifications — Overview” on page 5-1 if you are unfamiliar with local service.

Step 1. Prepare to Rework Local Service or Modifications

1 You need to prepare the system for service each time you rework service to your system. As a precaution, do a complete backup of your system. See the *VM/ESA: Installation Guide* for instructions on storing a backup copy of the VM/ESA system to tape.

2 Set up the correct virtual machine environment and IPL.

set machine xa
System reset.
System = XA

If your machine environment is already in the correct mode, you will not see all of these displayed messages.

define storage *nn***M**
STORAGE CLEARED - SYSTEM RESET
STORAGE = nnM

Define the storage to at least 35MB. If you will be servicing GCS, make sure to define your storage to be at least as large as the value you defined for HIGH COMMON END storage (default 19MB) during the installation procedures. For more information, see *VM/ESA: Installation Guide*.

Defining storage causes a system reset.

ipl 190 clear

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

ENTER

Press ENTER to complete the IPL.

3 Access the necessary VMSES/E disks:

access *address* **b**
access 51d d

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

4 If your PPF file has changed, enter the following command:

vmfppf esa *compname*

esa is the $PPF. If you have an override to **esa**, use it instead.

compname is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, CMS, CP, GCS, DV, TSAF, and AVS.

5 Set up your access order by entering the following command. Using the command output, make a note of the file modes of the LOCALMOD disk, the alternate APPLY disk, and the BASE disks for this component. The LOCALMOD disks and alternate APPLY disk must be accessed R/W.

vmfsetup esa *compname*

6 Rebuild source files.

If you have applied IBM supplied service and are applying local service, you need to insure that the source files are updated (in case the files were serviced by IBM). GCS does not ship source files so this step is not necessary for this component. For all other components you are processing, issue the following command.

vmfbld ppf esa *compname bldlist* **(serviced**

compame and *bldlist* are the names of the component and build list which can be:

for VMSES	VMFBLSRC
for REXX	IXXBLSRC
for CMS	DMSBLSRC
for CP	HCPBLSRC
for DV	HCSBLSRC
for TSAF	ATSBLSRC
for AVS	AGWBLSRC

Step 2. Rework Local Service to Source-Maintained Parts

Note

Complete Step 2 only if you are reworking service to a source-maintained part using update files. If you are reworking a change to a replacement-maintained part, skip to “Step 3. Rework Local Service to Replacement-Maintained Parts” on page 7-3.

1 XEDIT the file with the CTL option to determine if your local modification is still compatible with the IBM service.

a XEDIT the source file with the CTL option.

xedit *fn ft fm* **(ctl** *ctlfile*

fn, *ft* and *fm* are the file name, file type and file mode of the source file you are modifying.

b If you have sequence errors, then you need to fix the errors and reenter the XEDIT command. If you are placed in an XEDIT session then the sequencing is compatible. Make sure the logic of your local modification is compatible with the IBM service.

c Make any necessary changes to the displayed file, and enter the FILE command in XEDIT to place your changes in the update file.

====> file = = *fm-local*

2 Complete this procedure only if you are reworking a local modification to a source-maintained MACRO.

a Determine what MACLIBs need to be rebuilt.

vmfqobj esa *compname* **tdata :object** *fn* **:libname**

fn is the name of the MACRO that needs to be reworked.

Use VMFQOBJ to determine what MACLIBs need to be rebuilt.

:OBJECT *bldlist.fn*
:LIBNAME *libname*

bldlist is the name of a build list of a MACLIB that needs to be rebuilt. Record this build list name.

b Update the $SELECT file.

vmfrepl *fn ft* **esa** *compname* **($select nocopy filetype** *ft*

ft is the file type of the macro (for example, MACRO or COPY).

Step 3. Rework Local Service to Replacement-Maintained Parts

Note

Complete Step 3 only if you are reworking a change to a replacement-maintained part. If you are reworking a change to a source-maintained part you should have completed Step 2 and should now skip to “Step 4. Rebuild Objects” on page 7-4.

1 Determine the highest level of IBM service to the part.

vmfrepl *fn ft* **esa** *compname* **(query nocopy**

The highest level of IBM service to the part is returned in message VMFREP2511I.

If there is no IBM service to the part, there is no reason to rework the local service.

2 Compare the changes between the local modification and the IBM service. Determine how to include the IBM changes in your local modification.

- XEDIT the copy of the local modification on the LOCALMOD disk. Compare this to the contents of the highest level of IBM service.

xedit *fn ftabbrev-modid fm-local*

- Make your changes to your local modification.
- When you complete your changes, save the replacement part.

====> file

3 Update the $SELECT file for the part.

vmfrepl *fn ft* **esa** *compname* **($select nocopy**

4 If you are reworking a MACRO, use VMFQOBJ to determine what MACLIBs need to be rebuilt.

vmfqobj esa *compname* **tdata :object** *fn* **:libname**

:OBJECT *bldlist.fn*
:LIBNAME *libname*

fn is the name of the MACRO.

bldlist is the name of a build list of a MACLIB that needs to be rebuilt. Record this name for use in "Step 4d. Rebuild MACLIBs" on page 7-7.

Step 4. Rebuild Objects

Note

When you have completed the preceding steps for all the rework to the local modifications for this component, then rebuild the objects.

Step 4a. Rebuild Source Files

Note

Complete this step only if you have reworked a replacement part for a source file in "Step 3. Rework Local Service to Replacement-Maintained Parts" on page 7-3.

To rebuild the source for the component, issue the following command. (GCS does not ship source files so this step is not necessary for this component.)

vmfbld ppf esa *compname bldlist* **(serviced**

compame and *bldlist* are the names of the component and build list which can be:

VMSES	VMFBLSRC
REXX	IXXBLSRC
CMS	DMSBLSRC
CP	HCPBLSRC
DV	HCSBLSRC
TSAF	ATSBLSRC
AVS	AGWBLSRC

Step 4b. Create Replacement Part from Modified $Source, Compiled REXX

Note

Complete this step only if you have reworked a local modification to a compiled REXX part in “Step 2. Rework Local Service to Source-Maintained Parts” on page 7-2. You must complete substep 1 and either substep 2 or substep 3.

1 Create the uncompiled file.

vmfexupd *fn ft* **esa** *compname* **(outmode localmod $select ftabbr** *sftabbrev*

fn ft is the file name and file type (without the leading $ from the file type).

sftabbrev is **SXE** for EXEC files or **SXD** for XEDIT files.

This creates the updated file named *fn sftabbrev-modid* on *fm-local*, updates the $SELECT file and VVT.

2 Create the compiled file using the REXX compiler. If you do not have the REXX compiler, go to substep 3 on page 7-6.

a Use the REXXC command to create the compiled file.

rexxc *fn sftabbrev-modid fm-local* **(cexec(=** *cftabbrev-modid***)**

sftabbrev-modid is the file type of the output file from the previous step. It was identified in message VMFEXU2507I.

cftabbrev is the abbreviation for the compiled file type: **CEX** for EXEC files or **CAE** for XEDIT files.

b Update the $SELECT file. Skip substep 3 on page 7-6.

```
vmfrepl fn ft esa compname ($select ftabbr cftabbrev nocopy
```

3 Create a replacement for the compiled file using the VMFREPL command if you do not have the REXX compiler. This command also updates the $SELECT file.

```
vmfrepl fn ft esa compname = sftabbrev-modid fm-local ($select ftabbr cftabbrev outmode localmod
```

sftabbrev-modid is the file type of the output file from the previous step. It was identified in message VMFEXU2507I.

cftabbrev is the abbreviation for the compiled file type: **CEX** for EXEC files or **CAE** for XEDIT files.

Step 4c. Create Replacement Part from Modified $Source, Not Compiled REXX

Note

Complete this step only if you reworked a local modification to a $source file, **other than** a compiled REXX part, in Step 2.

1 Update the file with the VMFEXUPD command.

```
vmfexupd fn ft esa compname (outmode localmod $select
```

fn ft is the file name and file type (without the leading $ from the file type).

This creates the updated file named *fn ftabbrev-modid* on *fm-local* and updates the $SELECT file.

Notes:

a. The file type of the source file must begin with a dollar sign ($), but **do not enter the $ in the command**.

b. The VMFEXUPD command uses the options HISTORY and SID.

c. If the file to be updated is an EXEC or XEDIT macro that is included in the CMS installation saved segment (default name CMSINST), use the NOCOMMENTS option to remove all comments and leading blanks. Then, rebuild the CMS installation saved segment to include the updated file.

d. If you are using RECEIVEX $XEDIT, rename the updated RECEIVEX XEDIT output to RECEIVE XEDIT.

e. The file $DASD$ $CONSTS cannot use the general command format; therefore, use the following command with the additional option NOHIST:

```
vmfexupd $dasd$ consts esa cp (nohist outmode localmod $select
```

fn ft is the file name and file type (without the leading $ from the file type).

Step 4d. Rebuild MACLIBs

Note

Complete this step if you have reworked a local modification (update or replacement) to a macro or to a MACLIB build list in “Step 2. Rework Local Service to Source-Maintained Parts” on page 7-2 or “Step 3. Rework Local Service to Replacement-Maintained Parts” on page 7-3 or if you have IBM supplied service to a MACLIB.

For each MACLIB that needs to be rebuilt enter the following command.

Note: You only need to rebuild a MACLIB once even if it contains multiple MACROs that were changed.

vmfbld ppf esa *compname bldlist* **(serviced**

bldlist is the name of a build list of a MACLIB that needs to be rebuilt.

If for some reason the VMFBLD command is unsuccessful, rework the change to the MACRO.

Step 4e. Create a Replacement Part from Updated ASSEMBLE Files

Note

Complete this step only if you have reworked a local modification to an ASSEMBLE file in “Step 2. Rework Local Service to Source-Maintained Parts” on page 7-2, or if you have reworked a local modification to a macro in “Step 2. Rework Local Service to Source-Maintained Parts” on page 7-2 or “Step 3. Rework Local Service to Replacement-Maintained Parts” on page 7-3 and you have ASSEMBLE files that use the macro and must be reassembled.

For local modifications to the REXX/VM component, follow procedure 2; **for all other components,** follow procedure 1.

1 For components besides REXX/VM use the following command for ASSEMBLE files, except for the DMSTRT*cc* ASSEMBLE file. For the DMSTRT*cc* ASSEMBLE file, (where *cc* is a country code suffix), see “Step 4g. Create a Replacement Part from Modified National Language Files” on page 7-8. The command also updates the $SELECT file and places the output file on the LOCALMOD disk.

If you are assembling a CMS Pipelines part (any part with a file name of FPL*xxx* plus the following parts: DMSCRX, DMSPFP, DMSPPP, DMSPQI, DMSPRN, DMSXFB, MDATPEEK, and RITA), use the second form of the VMFHLASM command which includes the CNTRL option.

vmfhlasm *fn* **esa** *compname* **($select outmode localmod**

OR

vmfhlasm *fn* **esa cms ($select outmode localmod cntrl fplvm**

Go to "Step 4f. Create a Replacement Part from Replaced ASSEMBLE Files" for the next step.

2 REXX/VM text decks are included in the CMS nucleus. Because of this, you cannot use the component name of REXX. Use instead the component name of CMS with the VMFHLASM command to ensure the correct object naming.

a Issue the VMFHLASM command with CMS to ASSEMBLE the file for inclusion in the CMS nucleus. This also updates the $SELECT file and places the output file on the LOCALMOD disk.

vmfhlasm *fn* **esa cms ($select outmode localmod**

b In "Step 4h. Rebuild Remaining Objects" on page 7-9 you must rebuild CMS, but it is not necessary to rebuild REXX unless you have reworked a local modification to a REXX part other than an ASSEMBLE file.

Step 4f. Create a Replacement Part from Replaced ASSEMBLE Files

Note

Complete this step only if you have reworked a local modification to a replacement ASSEMBLE file in "Step 3. Rework Local Service to Replacement-Maintained Parts" on page 7-3.

1 Enter the VMFHLASM command to assemble the file. The command places the output file on the LOCALMOD disk and updates the $SELECT file.

vmfhlasm *fn* **esa** *compname* **(nockgen $select outmode localmod**

Step 4g. Create a Replacement Part from Modified National Language Files

Note

Complete this step only if you have local modifications to any of the following national language files:

- *fncc* REPOS file
- *fncc* DLCS file
- DMSTRT*cc* ASSEMBLE file (where *cc* is a country code suffix).

1 Use the VMFNLS EXEC to generate the text deck for each national language file. This also updates the $SELECT file and places the output file on the LOCALMOD disk.

vmfnls *fn ft* **esa** *compname* **($select outmode localmod**

fn and *ft* are the file name and file type of the source file.

For more information about the VMFNLS EXEC, see *VM/ESA: VMSES/E Introduction and Reference*.

Step 4h. Rebuild Remaining Objects

If you were sent here from Chapter 3, “Using the COR Service Procedure” on page 3-1, return to the next task in the procedure for servicing this component in Chapter 3, “Using the COR Service Procedure” on page 3-1.

Otherwise, if you were reworking a local modification, then you need to continue by building the rest of the objects affected by the rework. To do this, go to the build section for the component you are processing and complete that section. Use the following table to find the appropriate build section.

Table 7-1. Build Section Reference

Component	Build Section
VMSES/E	“Prepare to Build” on page 3-10
REXX/VM	“Prepare to Build” on page 3-17
CMS	“Prepare to Build” on page 3-24
CP	“Prepare to Build” on page 3-34
GCS	“Prepare to Build” on page 3-42
Dump Viewing Facility	“Prepare to Build” on page 3-51
TSAF	“Prepare to Build” on page 3-58
AVS	“Prepare to Build” on page 3-65

Chapter 8. Removing Service

You can remove (back out) service levels or individual PTFs by un-applying the appropriate PTFs using the VMFREM command, and rebuilding all objects that were affected by the service being removed. The VMFREM command can, optionally, also un-receive the un-applied PTFs; thus totally removing them from your system.

The following service levels are supported:

- The alternate (current) service level
- The intermediate (test) service level
- The production service level.

Each service level is located on a separate disk. For example, the access order specified in the default PPF for CP is:

APPLY (2A6)	Alternate apply disk
APPLY (2A4)	Intermediate apply disk
APPLY (2A2)	Production apply disk

The following sections identify the steps required to remove service levels and single PTFs.

Removing Service Levels

If you want to back out the current service level and use only the previous test service level, you must perform the following steps for one service level.

If you want to back out both the current and test levels and use only the production service level, you must perform the following steps for two service levels.

Attention

If you have done a merge (VMFMRDSK) on the component there is no service on the alternate apply disk, and therefore there is no current service to remove. For components that share disks, doing a merge on one component will merge all the files from the other component. For example, CMS and REXX share disks. If you merged the REXX files then you have also performed a merge on the CMS files since the components share the disks.

1 Access the service disks required for the component using the VMFSETUP command.

```
vmfsetup esa compname
```

2 Un-apply the current service level or un-apply the current and the test service levels.

vmfrem ppf esa *compname* **levels** *n*

n is the number of service levels to un-apply (*n*=1 for the current level, *n*=2 for the current and test levels)

If you get the following message during remove processing, you must decide how to proceed and then respond to message VMFREM2131R.

```
VMFREM2130W VMFREM has encountered the following
missing parts while attempting to remove service levels.
```

If you respond to continue, you must restore the missing parts before running VMFBLD. This can be done in one of the following ways:

- Retrieve them from a product, service, or backup tape.
- Generate them, if source maintained, by using VMFHLASM or VMFEXUPD.

3 Follow the build steps identified in Table 6-1 on page 6-14 for the component being processed. After all build steps have been completed for the component, all service that was on the specified levels will have been removed.

Removing A Single PTF

If you want to back out a single PTF, you must perform the following steps. Any PTFs that are dependent on the specified PTF will also be backed out.

1 Access the service disks required for the component using the VMFSETUP command.

vmfsetup esa *compname*

2 Un-apply the PTF and its dependents.

vmfrem ppf esa *compname* **ptf** *ptfnum*

ptfnum is the PTF number of the PTF to un-apply.

If you get the following message during remove processing, you must decide how to proceed and then respond to message VMFREM2131R.

```
VMFREM2130W VMFREM has encountered the following
missing parts while attempting to un-apply PTFs.
```

If you respond to continue, you must restore the missing parts before running VMFBLD. This can be done in one of the following ways:

- Retrieve them from a product, service, or backup tape.
- Generate them, if source maintained, by using VMFHLASM or VMFEXUPD.

3 Follow the build steps identified in Table 6-1 on page 6-14 for the component being processed. After all build steps have been completed for

the component, the specified PTF and its dependents will have been removed.

Restoring Removed Service

If you want to restore service that you removed using the previous procedures, you must perform the following steps:

1 Remove the PTFs that you want to restore from the exclude list. Each PTF that was un-applied by VMFREM was put in a specified exclude list. See message VMFREM2144I in the $VMFREM $MSGLOG for the file identifier of the exclude list. Edit the specified exclude list, and remove the PTFs to be restored.

Note

If the UNRECEIVE option was specified on the VMFREM command when the service was removed, the service must be re-received before the next step.

2 Re-apply the service.

vmfapply ppf esa *compname*

3 Follow the build steps identified in Table 6-1 on page 6-14 for the component being processed. After all build steps have been completed for the component, all service will have been restored.

Appendix A. Receiving Service for VMSES Envelopes

Note

Substitute these procedures for the *Receive the Service Documentation* procedure and the *Receive the Service* procedure in the steps for the component you are servicing.

Receive Service Documentation

1 Follow procedures in the *ServiceLink User's Guide* or the appropriate README file to retrieve and decompact the envelope file on your A-disk.

2 If the file name of the ServiceLink files are of the format VLST*nnnn* and VPTF*nnnn*, then they are in VMSES format. Use the following steps to receive the documentation and VM/ESA service. If the file name is of the format NLST*nnnn* and NPTF*nnnn*, then use the steps in Appendix B, “Receiving Service For Non-VMSES Envelopes” on page B-1 to receive the documentation and service instead.

3 Receive the service documentation from the documentation envelope VLST*nnnn* SERVLINK.

Notes:

a. You need to do this operation only once for each envelope file, no matter how many products or components are in each envelope file.

b. To create a SERVICE DISKMAP file for the corresponding PTF envelope, enter the following command and substitute VPTF*nnnn* for VLST*nnnn*.

`vmfrec info (env vlst`*`nnnn`*

nnnn is the last 4 digits of your COR order number.

VMFREC:

- Loads the COR VMELEDOC file which contains the order information for the envelope. This file is unique for each envelope. If there are multiple envelopes to be received, this file should be reviewed prior to receiving the service documentation from the next envelope.
- Loads the COR DOCUMENT.
- Loads the product memo files (*prodid* MEMO). *prodid* is the product ID of the product(s) or VM/ESA component(s) that are in the envelope.

4 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview receive

5 Read the product memo(s) (*prodid* MEMO) before going on. These memos are loaded to the C-disk if it is accessed R/W; otherwise, the memos are loaded to the A-disk which is accessed R/W in “Step 1. Prepare the System for Service” on page 3-5.

Receive the Service

1 Receive the service for VM/ESA from the PTF envelope VPTF*nnnn* SERVLINK.

vmfrec ppf esa *compname* **(env vptf***nnnn* **setup**

esa is the file name of the base product parameter file. If you have a product parameter override file, you should substitute that file name for **esa**. The file type must be $PPF.

compname is the name of the component.

VMFREC:

- Maps the envelope file (verifies that all the necessary files exist) and creates a SERVICE DISKMAP file.
- Loads service for the component.

2 Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before going on. For information about handling specific receive messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview receive

Use the PF10 key to look at all non-status messages.

Appendix B. Receiving Service For Non-VMSES Envelopes

Note

This appendix only describes how to receive service for non-VMSES envelopes. Because the service is in a non-VMSES format, the procedures in this book cannot be used for applying the service. Refer to the documentation that comes with the envelope.

1 Receive the product memos from the documentation envelope NLST*nnnn* SERVLINK.

vmfplcd rst env= nlst*nnnn* **servlink** *fm*

This command resets the logical position to the beginning of the envelope. The GLOBALV will also be updated with the name of the envelope so that the ENV= parameter will not be required for subsequent VMFPLCD commands for that envelope. *fm* is the file mode of the disk where the envelope exists. The default is the A-disk.

vmfplcd load * * a (eod

This command receives service documentation for the envelope (COR VMELEDOC, SERVICE DOCUMENT, and COR01 MEDIAMAP).

2 Read the memo-to-users for each product that you plan to service.

3 Receive the service for non-VMSES PTF envelopes NPTF*nnnn* SERVLINK.

vmfplcd rst env= nptf*nnnn* **servlink** *fm*

This command resets the logical position to the beginning of the envelope. The GLOBALV will also be updated with the name of the envelope so that the ENV= parameter will not have to be specified in subsequent VMFPLCD commands. *fm* is the file mode of the disk where the envelope exists. The default is the A-disk.

vmfplcd scan *prodid* **010101 (eod**

Locate the start of service in the envelope for the product to service.

prodid is the product ID for which you want to install service. The *prodid* can be obtained from the COR01 MEDIAMAP file, loaded in 1.

vmfplcd load * * *fm* **(disk**

Receive the service for the product on a work disk. Refer to the product documentation for procedures to follow for installing corrective service.

fm is the file mode of the work disk that you are using for the product you are installing.

Appendix C. Building the NLS Nuclei, Including UCENG

To change the default system language for CP, CMS or GCS you must rebuild the CP, CMS or GCS nuclei to contain the national language files for the language desired. The following table identifies the national languages supported by CP and CMS and the $PPF (Product Parameter File) used to build the new nuclei. Currently, GCS only supports UCENG.

If you are building the CMS nucleus, ensure that the DMSNGP ASSEMBLE file has the appropriate changes for the language you are building. See the *VM/ESA: Installation Guide* if changes are needed to the DMSNGP ASSEMBLE file. Also, to add an optional language as a system option, see the *VM/ESA: Installation Guide*.

Table C-1. National Language $PPF files

ppfname **$PPF File**	**National Language**
CANFR	Canadian French
FRANC	French
GER	German
KANJI	Kanji
UCENG	Uppercase English

1 Access the necessary VMSES/E disks:

```
access address b
access 51d d
```

address is 5E6 if you have serviced VMSES/E; otherwise, the *address* is 5E5.

2 Before building the CP, CMS or GCS nuclei, the PPF file must be created for the language. This is done using the VMFPPF command.

```
vmfppf ppfname cp|cms|gcs|gcssfs
```

Enter the VMFPPF command for CP, CMS or GCS.

3 If your access order has changed since you performed the last task, enter the following command:

```
vmfsetup ppfname cp|cms|gcs|gcssfs
```

After the PPF file is created, the CP, CMS or GCS nucleus needs to be rebuilt. In these procedures substitute the national language $PPF file name for **esa** or *ppfname*.

- To rebuild and IPL the CMS nucleus see “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15.
- To rebuild and save the CP load map see “Rebuild the CP Nucleus” on page 4-21 and “Verify the CP Load Map” on page 3-37.

- To rebuild and IPL the GCS nucleus see “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27.

Appendix D. Test the New Level of CMS

The VM Callable Services Library (VMLIB), Shared File System (SFS) modules, the REXX/VM component, and CMS nucleus are all functionally dependent on each other. Following is a minimal test of CMS to ensure that these functional dependencies are maintained after installing new service.

1 If you use the Shared File System, prepare an SFS server to be used for testing the new level of CMS. Otherwise, skip this task.

logon *userid*	Log on an SFS server user ID. VMSERVS is the recommended user ID, because it is an IBM-supplied server that is only used for service installation and maintenance.
stop	Stop the server. **Note:** To use the SFS modules on the test system tools disk and not from a saved segment, the NOSAVESEGID parameter must be specified in the server's start-up parameters (DMSPARMS) file. The DMSPARMS files for the IBM-supplied servers (VMSERVS, VMSERVR, and VMSERVU) specify SAVESEGID CMSFILES. This parameter must be changed to NOSAVESEGID.
xedit *userid* **dmsparms**	XEDIT the DMSPARMS file for the server.
====> top	Start at the top of the file.
====> set case mixed ignore	Insure that the search is not case significant.
====> locate/savesegid	Locate the SAVESEGID parameter.
====> change/SAVESEGID *savesegname*/**NOSAVESEGID**	Change the SAVESEGID *savesegname* parameter to NOSAVESEGID.
====> file	Save the updated DMSPARMS file.
define 190 590	Define the production CMS system disk as 590 to allow the user ID to link the test CMS system disk as 190.
define 193 593	Define the production system tools disk as 593 to allow the user ID to link the test system tools disk as 193.
link maint 490 190 rr	Link the test CMS system disk as 190.
link maint 493 193 rr	Link the test system tools disk as 193.
ipl 190 clear parm nosprof instseg no	IPL the new (test) level of CMS without executing the system profile (SYSPROF EXEC) (which loads the CMSPIPES saved segment) and without loading the installation saved segment (CMSINST).

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

***** DO NOT press ENTER! *****

access (noprof	Bypass the execution of the PROFILE EXEC.
access 193 b	Access the test system tools disk.
fileserv start	Start the server.
#cp disconnect	Disconnect from the SFS server user ID.

2 Prepare a user ID to be used for testing the new level of CMS.

logon *userid*	Log on the user ID used to test CMS. The MAINT user ID can be used, because it uses the VMSERVS server for its SFS directories.
define 190 590	Define the production CMS system disk as 590 to allow the user ID to link the test CMS system disk as 190.
define 193 593	Define the production system tools disk as 593 to allow the user ID to link the test system tools disk as 193.
link maint 490 190 rr	Link the test CMS system disk as 190.
link maint 493 193 rr	Link the test system tools disk as 193.
ipl 190 clear parm nosprof instseg no	IPL the new (test) level of CMS without executing the system profile (SYSPROF EXEC) (which loads the CMSPIPES saved segment) and without loading the installation saved segment (CMSINST).

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

3 Test the new level of CMS.

Test the new level of CMS using your installation's method. If you are using SFS, include tests using the SFS server prepared with the new (test) level of CMS. Minimally, this should include XEDITing and saving a file on an SFS directory.

4 Return the user ID used for testing to the old level of CMS.

detach 190	Detach the test CMS system disk.
detach 193	Detach the test system tools disk.
define 590 190	Define the production CMS system disk as 190.
define 593 193	Define the production system tools disk as 193.
ipl 190 clear	IPL the old (production) level of CMS.

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

disconnect	Disconnect from the user ID.

5 If the Shared File System was used in the test, return the SFS server to the old level of CMS. Otherwise, skip this task.

logon *userid*	Log on the SFS server user ID used to test the new level of CMS.
stop	Stop the server.
detach 190	Detach the test CMS system disk.
detach 193	Detach the test system tools disk.
define 590 190	Define the production CMS system disk as 190.
define 593 193	Define the production system tools disk as 193.
ipl 190 clear	IPL the old (production) level of CMS.

VM/ESA V2.4.0 *mm/dd/yy hh:mm*

***** DO NOT press ENTER! *****

access (noprof	Bypass the execution of the PROFILE EXEC.
access 193 b	Access the production system tools disk.
xedit *userid* **dmsparms**	XEDIT the DMSPARMS file for the server.
====> top	Start at the top of the file.
====> set case mixed ignore	Insure that the search is not case significant.
====> locate/nosavesegid	Locate the NOSAVESEGID parameter.
====> change/NOSAVESEGID/SAVESEGID *savesegname*	Change the NOSAVESEGID parameter to SAVESEGID *savesegname*. If you are using an IBM-supplied server (VMSERVS, VMSERVR or VMSERVU), *savesegname* is CMSFILES.
====> file	Save the updated DMSPARMS file.
fileserv start	Start the server.
#cp disconnect	Disconnect from the SFS server user ID.

6 Continue applying service from the original user ID which you are using for the service application.

logon *service_id*	*service_id* is the user ID which you are using to apply service.

Appendix E. Local Modification Example for CMSINST

CMS improves system performance and storage usage by placing heavily used EXECs in the CMS installation segment, CMSINST. If you want to add or delete an EXEC from CMSINST, you should identify the changes to VMSES/E using the local modification procedure. This will allow VMSES/E to track the changes and ensure the CMSINST segment is rebuilt when any of the EXECs in it are serviced.

CMSINST is a logical segment within the INSTSEG physical segment. Two files are involved in making changes to CMSINST:

- CMSINST LSEG lists the EXECs that are contained in CMSINST. This file must be changed to add or delete an EXEC from CMSINST.
- DMSSBINS EXEC is the CMSINST segment build list. It includes a :BLDREQ record for each of the EXECs in CMSINST that belong to the CMS component. This enables VMSES/E to indicate that CMSINST needs to be rebuilt when any of these EXECs are serviced. If you are adding or deleting an EXEC from CMSINST that belongs to the CMS component, you should add or delete a :BLDREQ record in DMSSBINS EXEC.

In this example, the CONV2WD EXEC is added to CMSINST. CONV2WD EXEC is in the DMSBLCRX build list in the CMS component.

1 Planning the changes to CMSINST.

If you are adding EXECs to CMSINST, you should ensure that the INSTSEG segment, which contains the CMSINST logical segment, is large enough for the additional EXECs. You can determine if there is enough space by looking at the entry for INSTSEG in the SYSTEM SEGID file. You can also look at the INSTSEG PSEGMAP, if it exists. Use the VMSES/E segment mapping tool, VMFSGMAP, to increase the range defined for INSTSEG if it is too small. For more information about this tool, see *VM/ESA: VMSES/E Introduction and Reference*.

Also, if an added EXEC is not on a disk accessed by the CMS component, you can use the :DISKS field in the INSTSEG entry in VMFSGMAP to cause the disk to be accessed before the INSTSEG segment is built. The disk must have already been linked by the USER ID which will build the segment.

a If changes are required to the INSTSEG segment invoke the segment mapping tool (VMFSGMAP) to change the segment definition.

```
vmfsgmap segbld esasegs segblist
```

This command will display a panel containing a map of the segments on your system. Place the cursor on the INSTSEG segment and select PF4 to display the change segment definition panel.

b Make the changes to the segment definition. From this panel the segment can be enlarged by changing the DEFPARMS field. Additional disks can be accessed by updating the :DISKS field. After making the changes select PF5 to return to the main map panel.

c File your changes by selecting PF5 from the main map panel if they are satisfactory.

2 Prepare for local service

a Establish the correct minidisk access order for CMS.

```
access 5e5 b
access 51d d
vmfsetup esa cms
```

Access the CMS disks since CMSINST LSEG and DMSSBINS EXEC are in CMS. Note the file modes of the LOCALMOD disk (file mode **E** in this example) and the alternate APPLY disk (**G**).

b Select a local modification id for this change

A modification id needs to be selected for this change. The *modid* should be unique to this local modification. In this example use **L0012** as the local *modid*.

3 Apply local service to CMSINST LSEG

The CMSINST LSEG is serviced by complete replacement; therefore, the highest existing level of IBM service needs to be found for this part. Modify the highest level part to create the new version.

a Copy the highest level of CMSINST LSEG to the LOCALMOD disk

```
vmfrepl cmsinst lseg esa cms ($select outmode localmod logmod l0012
```

segl0012 will be the file type of our replacement part for CMSINST LSEG. SEG is the file type abbreviation for LSEG and L0012 is the local *modid* that was selected for this change.

b Create the replacement part for CMSINST LSEG

Apply the changes to CMSINST SEGL0012 using XEDIT.

```
xedit cmsinst segl0012 e

====> input EXEC CONV2WD EXEC     *
====> file
```

XEDIT the replacement part and add the CONV2WD EXEC to it.

4 Apply local service to DMSSBINS EXEC

If an exec belongs to the CMS component, then DMSSBINS EXEC must be modified by adding or deleting a :BLDREQ record for each exec that is added to or deleted from CMSINST. If the exec does not belong to the CMS component then skip this step and continue to step 5. In the example, we need to add a :BLDREQ record for CONV2WD EXEC . This will ensure that when CONV2WD EXEC is serviced that the CMSINST saved segment will

also get built. DMSSBINS EXEC is source-maintained. The source file, DMSSBINS $EXEC, is located on one of the BASE disks.

a Create an AUX file entry for the modification.

```
xedit dmssbins auxlcl e (noprof
```

Update or create the DMSSBINS AUXLCL file on the LOCALMOD disk and add an entry for this update.

```
====> input ML0012DS LCL LCL0012   * add CONV2WD EXEC to CMSINST
====> file
```

Note: Notice that the same local *modid* **L0012** has been kept for this update since it is part of the same local modification.

b Create the update file associated with this change using XEDIT.

```
xedit dmssbins $exec p (ctl dmsvm
```

In this example, the BASE disk containing DMSSBINS $EXEC is file mode **P**. Note that DMSVM is the file name of the CMS control file.

```
====> locate /:BLDREQ./
```

Locate the beginning of the :BLDREQ records.

```
====> input          DMSBLCRX.CONV2WD.EXEC
```

Add the build requisite for CONV2WD EXEC. DMSBLCRX is the build list that contains CONV2WD EXEC.

```
====> file = = e
```

File the change to DMSSBINS $EXEC. The change now exists in the DMSSBINS ML0012DS file on the LOCALMOD disk.

c Create a replacement part for DMSSBINS EXEC.

Generate the new version of DMSSBINS EXEC. Since the DMSSBINS EXEC is created from a $EXEC source file, a replacement part needs to be created using VMFEXUPD. Later, VMFBLD will copy it to the appropriate BUILD disk. VMFEXUPD applies the local modification update created earlier to the source file DMSSBINS $EXEC. The output is named DMSSBINS EXCL0012 and is placed on the LOCALMOD disk. The $SELECT file will be updated with DMSSBINS EXC EXCL0003. (This is assuming that the highest previous level is EXCL0003.) The modification to DMSSBINS EXEC will be identified in 2VMVMA40 VVTLCL.

```
vmfexupd dmssbins exec esa cms ($select outmode localmod logmod
```

5 Rebuild the affected objects and place your local modification into production.

a First issue the following command to rebuild CMSINST LSEG and DMSSBINS EXEC.

```
vmfbld ppf esa cms dmssbins (serviced
```

b Rebuild the INSTSEG saved segment which contains the CMSINST logical saved segment. See “Rebuild the Saved Segments” on page 3-69.

Appendix F. Updating the CP Load List

This appendix shows how to update the CP load list based on the HCPMDLAT MACRO. If you are using an alternative macro see *IBM VM/ESA: CP Exit Customization* for how to update a *xxx*MDLAT MACRO.

The CP load list for VM/ESA, usually named CPLOAD EXEC, is an ordered listing of the modules in the CP Nucleus. This order is determined by the HCPMDLAT (Module Attribute) MACRO, which is called by the HCPLDL ASSEMBLE module.

The modules that make up the CP nucleus have one or more of these attributes:

- Resident modules. These modules are non-pageable.
- Pageable modules loaded at initialization. These modules may not be paged out until system initialization has finished.
- Other pageable modules. These modules are pageable in and out of real storage.
- Multiprocessor modules. These modules are capable of executing simultaneously on two or more processors.
- Non-Multiprocessor modules. These modules are only capable of executing on a single processor, the master processor, which is usually the IPL processor.

The CP load list must have fixed-length 80-byte records.

Within the HCPMDLAT MACRO the modules are listed in the order in which they will load at IPL. The modules are also grouped into various categories. If you are adding a new module to the CP load list, be sure to add it to the HCPMDLAT MACRO in the appropriate category, and in the appropriate order within that category. See the comments within the HCPMDLAT MACRO for more information.

There is another purpose for the HCPMDLAT MACRO besides producing the CP load list. When one CP module needs to call another, this MACRO gets the attribute information about the target entry point. Attributes of modules are:

- Pageable or Resident
- Multiprocessor eligible or Non-Multiprocessor eligible
- Savearea is either STATIC or DYNAMIC
- DATA module, as opposed to executable

Non-multiprocessor means that it will only be dispatched on the master processor.

If you are going to change the attributes of a module you must reassemble all the modules that call this module to pick up the new attributes. If you are making a module resident that was pageable, keep it within the same category, MP or non-MP, and add it to the end of the respective list.

If you need to update the CP load list, then use these instructions. This involves:

- Reviewing the guidelines for using update files,
- Creating a local modification for HCPMDLAT MACRO,
- Issuing the GENCPBLS command to create the CP load list,
- Updating the HCPGPI MACLIB, which contains HCPMDLAT MACRO, and
- Rebuilding the CP nucleus.

Steps to Follow

1 If you need to, review the recommended guidelines for using update files on page 6-4.

2 Modify HCPMDLAT MACRO using a local modification. Since HCPMDLAT MACRO is source-maintained, use the procedure for applying local service to source-maintained parts in "Step 3. Apply Local Service to Source-Maintained Parts" on page 6-3.

For the example in this appendix, the following variables will be used:

auxft	**AUXLCL**
compname	**CP**
cp_load_list	**CPLOAD**
fm-local	**E**
fn	**HCPMDLAT**
modid	**L0001**
ppfname	**ESA**
svclvl	**LCL**
updateft	**ML0001HP**

a Set up the correct environment. See "Step 1. Prepare for Local Service or Modifications" on page 6-1 for reference, but at a minimum you will need to access the CP disks with the VMFSETUP command. Note that the local disks and the alternate apply disk need to be accessed R/W.

```
vmfsetup esa cp
```

b Apply local service to HCPMDLAT MACRO (See "Step 3. Apply Local Service to Source-Maintained Parts" on page 6-3 for reference).

Update or create, on the local disk, the AUX file.

```
xedit hcpmdlat auxlcl e (noprof
====> input ML0001HP LCL LCL0001 * A change to CP Load List
====> file
```

c Create an update file.

- XEDIT the source file with the CTL option.

```
xedit hcpmdlat macro fm (ctl hcpvm
```

fm is the file mode of the source file you are modifying.

- Make your changes to the displayed source file. The original source file is **not** changed.
- When you complete your changes, save them on the LOCALMOD disk. When you enter the FILE command in XEDIT, all of your changes are placed in the update file, HCPMDLAT ML0001HP.

```
====> file = = e
```

3 Create the CP load list using GENCPBLS.

The local modification will be applied to HCPMDLAT MACRO. The output will be named CPLOAD EXCL0001. The file will be placed on the LOCALMOD disk. The 2VMVMB40 $SELECT file will be updated with two records (HCPMDLAT MACRO and CPLOAD EXC). The modification to HCPMDLAT MACRO and CPLOAD EXEC will be identified in 2VMVMB40 VVTLCL.

```
gencpbls cpload esa cp (outmode localmod $select logmod
```

4 Update the HCPGPI MACLIB.

The GENCPBLS command creates a temporary MACLIB with the updated copy of the HCPMDLAT MACRO. This allows you to determine if you have any assemble problems before the real MACLIB is updated. So, now you need to update HCPGPI, the MACLIB that contains HCPMDLAT.

```
vmfbld ppf esa cp hcpgpi (serviced
```

5 Rebuild the CP Nucleus.

To rebuild the CP nucleus and verify the CP load map, follow the instructions on “Rebuild the CP Nucleus” on page 4-21 and “Verify the CP Load Map” on page 4-22.

Appendix G. Using VMFBLD Outside of the Service Process

If you need to build an object or nucleus outside of the service process, perform the following steps.

1 If necessary, generate the appropriate level of PPF. You only need to do this if you have modified your PPF override file.

vmfppf *ppfname compname*

2 Access the disks.

vmfsetup *ppfname compname*

3 Locate the object to be built in one of the build lists. VMFBLD requires that you enter the name of the build list on the VMFBLD command.

vmfqobj *ppfname compname* **tdata :object** *objname*

You can use **:partid** *fn* instead of **:object** *objname* if you do not know the name of the object but you know the name of the part.

4 Build the object or nucleus.

a If the build list is one of the following:

CMSLOAD then rebuild and IPL the CMS nucleus. See “Rebuild the CMS Nucleus” on page 4-13 and “IPL the CMS Nucleus” on page 4-15.

CPLOAD then rebuild and save the CP load map. See “Rebuild the CP Nucleus” on page 4-21 and “Verify the CP Load Map” on page 4-22.

GCTLOAD then rebuild and IPL the GCS nucleus. See “Rebuild the GCS Nucleus” on page 4-26 and “IPL the GCS Nucleus” on page 4-27.

b Otherwise, enter the VMFBLD command specifying the build list, the name of the object and the ALL option.

vmfbld ppf *ppfname compname bldlist objname* **(all**

If the object just built resides in a segment then you will need to issue the VMFBLD command again to rebuild the build list and then rebuild the segment. To determine if a saved segment is affected review the build message log ($VMFBLD $MSGLOG) for a build requirement for the segment's build list (*segment_bldlist*). If necessary, enter the following command:

vmfbld ppf *ppfname compname segment_bldlist* **(serviced**

Then to rebuild the segment, see the appropriate VMFBLD command for the segment on “Rebuild the Saved Segments” on page 3-69.

Appendix H. Apply Changes Directly to Object Code

There are two methods to apply changes to object code. The first method is by using the ZAP or ZAPTEXT command. This is the preferred method. The second method, creating and applying a patch, is not the preferred method. The patch method does not allow automatic local notification from VMFAPPLY processing. If for some reason you need to use this method, see Appendix I, "Applying and Removing Patches" on page I-1.

A zap is a direct change to object code. There are two types of zaps:

- The ZAP command changes MODULE, TXTLIB, and LOADLIB files. The changes or instructions are supplied to ZAP through input control records.
- The ZAPTEXT EXEC changes individual text decks. ZAPTEXT uses the same input control records that ZAP uses, as well as the EXPAND control record.

Note

You should only use the ZAP command to change MODULEs, TXTLIBs, or LOADLIBs if they are serviced by complete replacement. If a MODULE, TXTLIB, or LOADLIB is built from serviced TEXT decks, you should use the ZAPTEXT command to change the appropriate TEXT decks.

Note

ZAP and ZAPTEXT changes cannot be removed from the modified text deck.

1 Use the VMFREPL command to copy the highest level of the part to the A-disk.

vmfrepl *fn ft ppfname compname* **(filetype** *ft*

fn and *ft* are the real file name and file type of the part to be modified.

2 Create the control records and change the object code.

For Text Decks

Create the ZAPTEXT input control records to define your changes. For more information about creating these records, see *VM/ESA: VMSES/E Introduction and Reference*.

Enter the ZAPTEXT command. For information about the format of this command, see *VM/ESA: CMS Command Reference*.

For MODULEs, TXTLIBs and LOADLIBs

Create the ZAP input control records to define your changes. For more information about creating these records, see *VM/ESA: VMSES/E Introduction and Reference*.

Enter the ZAP command. For information about the format of this command, see *VM/ESA: VMSES/E Introduction and Reference*.

3 Use VMFREPL to copy the modified object to the local disk, update the local version vector table, and update the $SELECT file.

vmfrepl *fn ft ppfname compname* **= = a ($select logmod** *modid* **outmode** *fm-local*

modid is the new local modification identifier for the part.

fm-local is the file mode of the local disk or the symbolic name for the local disk in the :MDA section of the PPF (for example, LOCALMOD for the ESA PPF).

4 Erase the modified part from the A-disk.

erase *fn ft* **a**

5 Enter the VMFBLD command with the SERVICED option to build the appropriate objects.

vmfbld ppf *ppfname compname* **(serviced**

Appendix I. Applying and Removing Patches

There are two methods to apply changes to object code. The first method is by using the ZAP or ZAPTEXT command. This is the preferred method. For instructions see Appendix H, “Apply Changes Directly to Object Code” on page H-1. The second method, creating and applying a patch, is not the preferred method. The patch cannot be logged in the VVT because it is processed dynamically when the text file is loaded so there is no replacement part generated. Thus this method does not allow automatic local notification from VMFAPPLY processing. If for some reason you need to use this method, use this appendix.

A patch is an update to object code contained in a text deck. The text deck can be used to build a module or can be included in a nucleus. The patch is applied using the service control structure and is nondestructive. That is, the patch does not change the text deck and can be backed out.

Creating a Patch

Use the following procedure to set up the update control structure, create a patch update file, and apply the patch.

1 Set up the control file to include the patch.

XEDIT the control file for the component you are changing and add a patch ID (TX$) to the entry for the AUX file you want to use to identify the patch update file. The following example shows a patch ID that has been added to the entry for the AUXPAT AUX file.

```
TEXT MACS   HCPGPI HCPPSI HCPOM1 HCPOM2 DMSGPI DMSOM OSMACRO
PAT  AUXPAT TX$  * LOCAL PATCHES
LCL  AUXLCL
TEXT AUXVM
```

Figure I-1. Sample Control File with Patch Identifier

2 Set up the AUX file to include the patch.

XEDIT the AUX file and add an entry for the patch update file. The file name of the AUX file must match the file name of the text deck to be patched. The file type of the AUX file must match the AUX identifier from the control file.

To patch module HCPABC using the control file shown above, the AUX file would be named would be HCPABC AUXPAT. The following example shows the patch entry in the AUX file.

```
VM12345 TX$ LCL2345 * comment
```

In this example, VM12345 (the APAR number) is the file type of a patch update file supplied by IBM. If the update file is not supplied by IBM, you can use any non-restricted file type.

3 Create the patch update file.

This file might be supplied by IBM.

a Create a patch update file. The file name of this file must match the file name of the text deck to be patched. The file type of the file is the identifier that you used as the first token in the patch entry in the AUX file. In our example, the file ID is HCPABC VM12345.

b Add the patch statements given to you by your IBM software support representative to the patch update file you just created.

4 Copy the patch update file onto the LOCALMOD disk.

5 Apply the patch.

Rebuild the affected nucleus or module. Use the procedure in the build step in Chapter 4, “Detailed Component Build Steps” on page 4-1 for the affected component. When you enter the VMFBLD command use the option of **ALL** instead of **SERVICED**.

Removing a Patch

To remove a patch:

1 XEDIT the AUX file that contains the patch and comment out the patch entry. Add a comment in the AUX file that explains why you are removing the patch. This comment is added to the nucleus map.

2 Rebuild the nucleus or module. Use the procedure in the build step in Chapter 4, “Detailed Component Build Steps” on page 4-1 for the affected component. When you enter the VMFBLD command use the option of **ALL** instead of **SERVICED**.

Appendix J. VM/ESA Build Lists

The following table identifies the build lists used in servicing VM/ESA and describes their functions. You can view the contents of each build list using XEDIT.

Table J-1 (Page 1 of 5). VM/ESA Build Lists

Build List	Component	Target	Description/Function
VMFBLHLP	VMSES/E	19D	This file is input to VMFBLD to copy VMSES/E replacement help files to the help disk.
VMFBLNON	VMSES/E	493/193	This file is input to VMFBLD to copy non-VMSES/E replacement parts to the system disk for general users.
VMFBLSES	VMSES/E	5E6/5E5	This file is input to VMFBLD to copy VMSES/E replacement parts to the VMSES/E installation and service tools disk.
VMFBLSRC	VMSES/E	493/193	This file is input to VMFBLD to copy VMSES/E replacement source files to the system tools disk.
VMFBLSYS	VMSES/E	490/190	This file is input to VMFBLD to copy VMSES/E replacement parts to the system disk for general users.
VMFMLOAD	VMSES/E	5E6/5E5	This file is input to VMFBLD to build VMSES/E modules on the VMSES/E installation and service tools disk.
VMFSBHLP	VMSES/E	N/A	This file is input to VMFBLD to update the build status for the help segment.
VMFSLOAD	VMSES/E	490/190	This file is input to VMFBLD to build VMSES/E modules on the system disk for general users.
IXXBLNUC	REXX/VM	493/193	This file is input to VMFBLD to force a message that the CMS and GCS nuclei must be rebuilt whenever a REXX/VM text deck has been serviced.
IXXBLSRC	REXX/VM	493/193	This file is input to VMFBLD to copy REXX/VM replacement source files to the system tools disk.
IXXOM	REXX/VM	490/190	This file is input to VMFBLD to build the IXXOM MACLIB.
IXXRP	REXX/VM	493/193	This file is input to VMFBLD to build the IXXRP MACLIB.
AMQOM	CMS	490/190	This file is input to VMFBLD to build the AMQOM MACLIB.
ASDREUSP	CMS	493/193	This file is input to VMFBLD to build the ASDREUSP MACLIB.
CMSMLOAD	CMS	490/190	This file is input to VMFBLD to build CMS modules on the system disk.
CMSLOAD	CMS	490/190	This file is input to VMFBLD to build the CMS nucleus.
CMSSAA	CMS	493/193	This file is input to VMFBLD to build the CMSSAA TXTLIB.
CMSXLOAD	CMS	493/193	This file is input to VMFBLD to build CMS modules on the system tools disk.
DMSBL400	CMS	400	This file is input to VMFBLD to copy replacement files to the workstation and BFS disk.
DMSBL490	CMS	490/190	This file is input to VMFBLD to copy CMS replacement files to the system disk.
DMSBL493	CMS	493/193	This file is input to VMFBLD to copy CMS replacement files to the system tools disk.
DMSBLAEN	CMS	493/193	This file is input to VMFBLD to DMSAENV TXTLIB.
DMSBLAMT	CMS	493/193	This file is input to VMFBLD to DMSAMT TXTLIB.

Table J-1 (Page 2 of 5). VM/ESA Build Lists

Build List	Component	Target	Description/Function
DMSBLASM	CMS	493/193	This file is input to VMFBLD to copy replacement text decks for the F assembler to the system tools disk.
DMSBLBKW	CMS	493/193	This file is input to VMFBLD to build the BKWLIB TXTLIB.
DMSBLCEN	CMS	493/193	This file is input to VMFBLD to build the DMSCENV TXTLIB.
DMSBLCMT	CMS	493/193	This file is input to VMFBLD to build the DMSCMT TXTLIB.
DMSBLCRX	CMS	490/190	This file is input to VMFBLD for the compiled REXX/VM files.
DMSBLDOS	CMS	493/193	This file is input to VMFBLD to copy replacement text decks for the DOS segment to the system tools disk.
DMSBLFAS	CMS	490/190	This file is input to VMFBLD to copy replacement text decks for the assembler to the system tools disk.
DMSBLHLP	CMS	19D	This file is input to VMFBLD to copy CMS replacement help files to the help disk.
DMSBLINF	CMS	493/193	This file is input to VMFBLD to copy CMS replacement text decks OS LOADed by the Dump Viewing Facility to the system tools disk.
DMSBLJAV	BFS	N/A	This file is input to VMFBLD to move JAVA files into the byte file system.
DMSBLNRX	BFS	N/A	This file is input to VMFBLD to move NETREXX files into the byte file system.
DMSBLNUH	CMS	19D	This file is input to VMFBLD to copy CMS replacement HELP files, that cannot be uppercased, to the HELP desk.
DMSBLPRP	CMS	490/190	This file is input to VMFBLD to build the PROPLIB LOADLIB.
DMSBLRAS	CMS	493/193	This file is input to VMFBLD to copy the sample RAS tools to the 493 disk.
DMSBLSRC	CMS	493/193	This file is input to VMFBLD to copy CMS replacement source files to the system tools disk.
DMSBLVML	CMS	490/190	This file is input to VMFBLD to build the VMLIB CSLLIB and VMLIB CSLSEG.
DMSBLVMT	CMS	490/190	This file is input to VMFBLD to build the VMMTLIB CSLSEG.
DMSGPI	CMS	490/190	This file is input to VMFBLD to build the DMSGPI MACLIB.
DMSOM	CMS	490/190	This file is input to VMFBLD to build the DMSOM MACLIB.
DMSRP	CMS	493/193	This file is input to VMFBLD to build the DMSRP MACLIB.
DMSSBBAM	CMS	N/A	This file is input to VMFBLD to update the build status for the CMSBAM segment.
DMSSBDOS	CMS	N/A	This file is input to VMFBLD to update the build status for the CMSDOS segment.
DMSSBCAN	CMS	N/A	This file is input to VMFBLD to update the build status for the Canadian French segment.
DMSSBFRA	CMS	N/A	This file is input to VMFBLD to update the build status for the French segment.
DMSSBGER	CMS	N/A	This file is input to VMFBLD to update the build status for the German segment.
DMSSBHLP	CMS	N/A	This file is input to VMFBLD to update the build status for the Help segment.

Table J-1 (Page 3 of 5). VM/ESA Build Lists

Build List	Component	Target	Description/Function
DMSSBINS	CMS	N/A	This file is input to VMFBLD to update the build status for the CMSINST segment.
DMSSBKAN	CMS	N/A	This file is input to VMFBLD to update the build status for the Kanji segment.
DMSSBPIP	CMS	N/A	This file is input to VMFBLD to update the build status for the CMSPIPES segment.
DMSSBSFS	CMS	N/A	This file is input to VMFBLD to update the build status for the CMSFILES segment.
DMSSBSVM	CMS	N/A	This file is input to VMFBLD to update the build status for the SVM segment.
DMSSBUCE	CMS	N/A	This file is input to VMFBLD to update the build status for the Uppercase English segment.
DMSSBVML	CMS	490/190	This file is input to VMFBLD to update the build status for the VMLIB segment.
FPLGPI	CMS	490/190	This file is input to VMFBLD to build the FPLGPI MACLIB.
FPLOM	CMS	490/190	This file is input to VMFBLD to build the FPLOM MACLIB.
GUIBLCSL	CMS	490/190	This file is input to VMFBLD to build the GUICSLIB CSLSEG.
GUIBLEXS	CMS	490/190	This file is input to VMFBLD to copy GUI mods segment files.
GUISBCSL	CMS	N/A	This file is input to VMFBLD to update the build status for the GUICSLIB segment.
GUISBDSK	CMS	N/A	This file is input to VMFBLD to update the build status for the GUIVMGUI segment.
IOCPLOAD	CMS	490/190	This file is input to VMFBLD to build IOCP modules on the system disk.
MVSXA	CMS	490/190	This file is input to VMFBLD to build the MVSXA MACLIB.
OSMACRO	CMS	490/190	This file is input to VMFBLD to build the OSMACRO MACLIB.
OSMACRO1	CMS	490/190	This file is input to VMFBLD to build the OSMACRO1 MACLIB.
OSPSI	CMS	490/190	This file is input to VMFBLD to build the OSPSI MACLIB.
SERVLOAD	CMS	493/193	This file is input to VMFBLD to build SFS server modules on the system tools disk.
CPLOAD	CP	493/193	This file is input to VMFBLD to build the CP nucleus.
HCPBL490	CP	490/190	This file is input to VMFBLD to copy CP replacement files to the system disk.
HCPBL493	CP	493/193	This file is input to VMFBLD to copy CP replacement files to the system tools disk.
HCPBLHLP	CP	19D	This file is input to VMFBLD to copy CP replacement help files to the help disk.
HCPBLIMG	CP	493/193	This file is input to VMFBLD to copy CP replacement GENIMAGE files to the system tools disk.
HCPBLINS	CP	493/193	This file is input to VMFBLD to copy replacement installation files to the system tools disk.
HCPBLRAS	CP	493/193	This file is input to VMFBLD to copy CP dump tool files to the system tools disk.

Table J-1 (Page 4 of 5). VM/ESA Build Lists

Build List	Component	Target	Description/Function
HCPBLSAM	CP	493/193	This file is input to VMFBLD to copy CP replacement sample files to the system tools disk.
HCPBLSDK	CP	490/190	This file is input to VMFBLD to copy CP replacement text decks to the system disk.
HCPBLSRC	CP	493/193	This file is input to VMFBLD to copy CP replacement source files to the system tools disk.
HCPBLUTL	CP	490/190	This file is input to VMFBLD to build CP utilities on the system disk.
HCPGPI	CP	490/190	This file is input to VMFBLD to build the HCPGPI MACLIB.
HCPMLOAD	CP	490/190	This file is input to VMFBLD to build CP modules on the system disk.
HCPOM1	CP	493/193	This file is input to VMFBLD to build the HCPOM1 MACLIB.
HCPOM2	CP	493/193	This file is input to VMFBLD to force build the HCPOM2 MACLIB.
HCPPSI	CP	490/190	This file is input to VMFBLD to build the HCPPSI MACLIB.
HCPRP	CP	493/193	This file is input to VMFBLD to build the HCPRP MACLIB.
HCPXLOAD	CP	493/193	This file is input to VMFBLD to build CP modules on the system tools disk.
HCPSBCAN	CP	N/A	This file is input to VMFBLD to update the build status for the Canadian French segment.
HCPSBFRA	CP	N/A	This file is input to VMFBLD to update the build status for the French segment.
HCPSBGER	CP	N/A	This file is input to VMFBLD to update the build status for the German segment.
HCPSBHLP	CP	N/A	This file is input to VMFBLD to update the build status for the Help segment.
HCPSBKAN	CP	N/A	This file is input to VMFBLD to update the build status for the Kanji segment.
HCPSBUCE	CP	N/A	This file is input to VMFBLD to update the build status for the Uppercase English segment.
GCTBLHLP	GCS	19D	This file is input to VMFBLD to copy GCS replacement help files to the help disk.
GCTBLINF	GCS	493/193	This file is input to VMFBLD to copy GCS replacement text decks OS LOADed by the Dump Viewing Facility to the system tools disk.
GCTBLPRP	GCS	595/595	This file is input to VMFBLD to build the PROPMX LOADLIB.
GCTBLRAS	GCS	493/193	This file is input to VMFBLD to copy GCS serviceability tools to the system tools disk.
GCTBLRST	GCS	493/193	This file is input to VMFBLD to copy GCS replacement parts to the system tools disk.
GCTGPI	GCS	493/193	This file is input to VMFBLD to build the GCTGPI MACLIB.
GCTLOAD	GCS	493/193	This file is input to VMFBLD to build the GCS nucleus.
GCTSBHLP	GCS	N/A	This file is input to VMFBLD to update the build status for the help segment.

Table J-1 (Page 5 of 5). VM/ESA Build Lists

Build List	Component	Target	Description/Function
HCSBL493	Dump Viewing Facility	493/193	This file is input to VMFBLD to copy Dump Viewing Facility replacement files to the system tools disk.
HCSBLHLP	Dump Viewing Facility	19D	This file is input to VMFBLD to copy Dump Viewing Facility replacement help files to the help disk.
HCSBLMSG	Dump Viewing Facility	493/193	This file is input to VMFBLD to copy Dump Viewing Facility replacement text decks to the system disk.
HCSBLSRC	Dump Viewing Facility	493/193	This file is input to VMFBLD to copy Dump Viewing Facility replacement source files to the system tools disk.
HCSMLOAD	Dump Viewing Facility	493/193	This file is input to VMFBLD to build Dump Viewing Facility modules on the tools disk.
HCSSLOAD	Dump Viewing Facility	490/190	This file is input to VMFBLD to build Dump Viewing Facility modules on the system disk.
HCSSBHLP	Dump Viewing Facility	N/A	This file is input to VMFBLD to update the build status for the help segment.
ATSBLHLP	TSAF	19D	This file is input to VMFBLD to copy TSAF replacement help files to the help disk.
ATSBLINF	TSAF	493/193	This file is input to VMFBLD to copy TSAF replacement text decks OS LOADed by the Dump Viewing Facility to the system tools disk.
ATSBLMES	TSAF	493/193	This file is input to VMFBLD to copy TSAF replacement message text decks to the system tools disk.
ATSBLSRC	TSAF	493/193	This file is input to VMFBLD to copy TSAF replacement source files to the system tools disk.
ATSMLOAD	TSAF	493/193	This file is input to VMFBLD to build the TSAF module on the system tools disk.
ATSSBHLP	TSAF	N/A	This file is input to VMFBLD to update the build status for the help segment.
AGWBLHLP	AVS	19D	This file is input to VMFBLD to copy AVS replacement help files to the help disk.
AGWBLINF	AVS	493/193	This file is input to VMFBLD to copy AVS replacement text decks OS LOADed by the Dump Viewing Facility to the system tools disk.
AGWBLLIB	AVS	493/193	This file is input to VMFBLD to build the AGW LOADLIB.
AGWBLSRC	AVS	493/193	This file is input to VMFBLD to copy AVS replacement source files to the system tools disk.
AGWBLUTL	AVS	493/193	This file is input to VMFBLD to rebuild the AGWUTIL load library to the system tools disk.
AGWSBHLP	AVS	493/193	This file is input to VMFBLD to update the build status for the help segment.

Appendix K. Determine Component Service Level

In addition, CMS, CP and GCS also provide commands to query the RSU service level for that component.

Determine CMS Service Level

In the example, 901 represents the RSU service level (9901).

```
query cmslevel
CMS Level 15, service level 901
```

Determine CP Service Level

In the example, 9901 is the RSU service level.

```
query cplevel
VM/ESA Version 2 Release 4.0, service level 9901
Generated at 04/03/98 12:11:53 EST
IPL at 04/03/98 03:12:09 EST
```

Determine GCS Service Level

In the example, 901 represents the RSU service level (9901).

```
query gcslevel
VM/ESA Version 2 Release 4.0, Service Level 901
```

Glossary

The list is also available in the online VM/ESA HELP facility. For example, to display the definition of “cms,” enter:

```
help glossary cms
```

You will enter the HELP facility's online glossary file and the definition of “cms” will be displayed as the current line. When you are in the glossary file, you can also search for other terms.

If you are unfamiliar with the HELP facility, you can enter:

```
help
```

to display the main HELP menu, or enter:

```
help cms help
```

for information about the HELP command.

For more information about the HELP facility, see *VM/ESA: CMS User's Guide*; for more about the HELP command, see *VM/ESA: CMS Command Reference*.

You can find additional information about IBM terminology in *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A

access mode. A method VM/ESA uses to control user access to data files. Access modes let the user read and write data to a file, or only read data from a file. See *file mode*.

alias. A pointer to an SFS base file. An alias can be in the same directory as the base file or in a different directory. There must always be a base file for the alias to point to. The alias references the same data as the base file. Data is not moved or duplicated.

alphanumeric. A character set that contains letters, digits, and usually other characters, such as punctuation marks.

APAR. Authorized program analysis report.

APAR number. The number that IBM assigns to an APAR and to the change resulting from it.

application program. A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll.

APPLIED. This status, listed in the apply status table, indicates a product or program temporary fix has been APPLIED to the system.

apply. When servicing a product or component, to generate an auxiliary control structure from a PTF.

Apply disk. In VMSES/E, a minidisk or SFS directory containing the files that describe the maintenance levels: the apply status table, AUX files, version vector tables, the select data file, and the build status table.

apply list. A file listing PTFs applied to a product or component.

apply status table. The Software Inventory table that identifies what PTFs have been applied to the product. The system level of the table identifies what product or component has been applied to the system. The file type of the system level inventory table is SYSAPPS and the file type of the service level inventory table is SRVAPPS.

Apply string. In VMSES/E, the set of Apply disks.

area. A term acceptable for DASD space when there is no need to differentiate between space on count-key-data devices and FB-512 devices. See *DASD space*.

assembler language. A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with instruction formats and data formats of the computer.

authority. In SFS, the permission to access a file or directory. You can have read authority or write authority (which includes read authority). You can also have file pool administration authority, which is the highest level of authority in a file pool.

authorized program analysis report (APAR). An official request to the responsible IBM Change Team to look into a suspected problem with IBM code or documentation. APARs describe problems giving conditions of failure, error messages, abend codes, or other identifiers. They also contain a problem summary and resolution when applicable. See *program temporary fix (PTF)*.

AUX file. Auxiliary control file.

auxiliary control file (AUX file). A file that contains a list of file types of update files applied to a particular source file or to control the service level used during build. See *control file* and *preferred auxiliary file*. Synonymous with *auxiliary file*.

auxiliary file. Synonym for *auxiliary control file*.

AVS. APPC/VM VTAM Support.

B

Base disk. In VMSES/E, a minidisk or SFS directory containing the original product code.

base file. The first occurrence of an SFS file. It remains the base for the life of the file, even if the file has been renamed. Aliases point to base files.

base file type. In VMSES/E, the file type used for a serviceable part when there is no service. The PTF number in the file type is set to “00000.” For example, EXC00000 would be the base file type for an exec. See serviceable part.

Base string. IN VMSES/E, the set of Base disks.

block. A unit of DASD space on FB-512 devices. For example, FB-512 devices can be the IBM 9335, 9332, 9313, 3370, and 3310 DASD using fixed-block architecture. In CMS Multitasking, to stop the execution of a thread until a function has been completed or a condition is satisfied.

Bpi. Bytes per inch.

bpi. Bits per inch.

build. In the installation and service of a product, to do the necessary steps to produce executable code or systems. This is often called the *build process*.

BUILDALL. This status, shown in the service-level build status table, indicates the user requested that an object be built with the ALL option on the VMFBLD command, and the object still needs to be built.

Build disk. In VMSES/E, a minidisk or SFS directory containing the running code for the product being serviced.

Build ID. A 1- to 8- alphanumeric character identifier (*bldid*) that is used to name the Software Inventory files created during build processing. The user can change this value to define different maintenance levels.

build list. An EXEC file that names the parts included in an object being built.

build requisites. An object that is needed to build another object. For example, when one object is built using another object, the latter is a build requisite of the former. Also, if an object's build requisite is serviced, the object must be rebuilt after its build requisite is built.

build status table. The Software Inventory table that identifies what products have been built, in the system level, and what individual objects have been generated for the product, in the service level. The file type of the system level inventory table is SYSBLDS and the file type of the service level inventory table is SRVBLDS.

Build string. The set of Build disks.

build-time requisites. Product(s) that must be installed before a certain product can run correctly.

BUILT. This status, listed in the build status table, indicates that a product or object has been built on the system.

C

callable services library (CSL). A package of CMS assembler routines that can be stored as an entity and made available to a high-level language, REXX, or an assembler program.

changes. In installation and service, service supplied by IBM and original equipment manufacturers (OEMs) for their programs. In the IBM service process, there are many ways users can receive information they need to fix (change) a portion(s) of a product they are running on a VM system. These include PTFs, APARs, user modifications, and information received over the phone. All these types of information are called *changes*.

checkpoint (CKPT) start. A VM/ESA system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. The spool file chains are reconstructed, but the original sequence of spool files is lost. Unlike warm start, CP accounting and system message information is also lost. Contrast with *cold start, force start,* and *warm start*.

circumventive service. Information that IBM supplies over the phone or on a tape to circumvent a problem by disabling a failing function until a PTF is available to be shipped as a corrective service fix. See *patch* and *zap*.

CKD. Count-key-data.

class A user. See *primary system operator privilege class.*

class authority. Privilege assigned to a virtual machine user in the user's directory entry; each class specified allows access to a subset of all the CP commands. See *privilege class* and *user class restructure (UCR)*.

CMS. Conversational Monitor System.

CMS EXEC. An EXEC procedure or EDIT macro written in the CMS EXEC language and processed by the CMS EXEC processor. Synonymous with *CMS program*.

CMS EXEC language. A general-purpose, high-level programming language, particularly suitable for EXEC procedures and EDIT macros. The CMS EXEC processor executes procedures and macros (programs) written in this language. Contrast with *EXEC 2 language* and *Restructured Extended Executor (REXX) language*.

CMS minidisk file directory. A directory on each CMS disk that contains the name, format, size, and location of each of the CMS files on that disk. When a disk is accessed by the ACCESS command, its directory is read into virtual storage and identified with any letter from A through Z. Synonymous with *master file directory block* and *minidisk directory*.

CMS nucleus. The portion of CMS that is resident in the user's virtual storage whenever CMS is executing. Each CMS user receives a copy of the CMS nucleus when the user IPLs CMS. See *saved system* and *shared segment*.

CNTRL file. Control file with file type CNTRL.

cold start. A VM/ESA system restart that ignores previous data areas and accounting information in main storage, and the contents of paging and spool files on CP-owned disks. Contrast with *checkpoint (CKPT) start, force start,* and *warm start*.

command. A request from a user at a terminal for the execution of a particular CP, CMS, GCS, TSAF, Dump Viewing Facility, or AVS function. A CMS command can also be the name of a CMS file with a file type of EXEC or MODULE. See *subcommand* and *user-written CMS command*.

command line. The line at the bottom of display panels that lets a user enter commands or panel selections. It is prefixed by an arrow (====>).

commit. In the context of SFS, to change a resource (such as a file) permanently. In the context of CRR, to make permanent changes to protected resources (such as SFS file pools) during a transaction (CRR logical unit of work). CRR commits changes made by an application program or transaction program.

COMMITTED. This status, listed in the receive status table, indicates that a PTF has been committed for the product. This means that obsolete parts of the PTF may be discarded.

common storage. A shared segment of reentrant code that contains free storage space, the GCS supervisor, control blocks, and data that all members of a virtual machine group share.

compile. To translate a program written in a high-level programming language into a machine language program.

component. A collection of objects that together form a separate functional unit. A product may contain many components. For example, CP, CMS, and TSAF are components of VM/ESA.

component override. Synonym for *component parameter override*.

component override area. An area of the product parameter file or of a product parameter override file that contains one or more component parameter overrides. Synonymous with *override area*.

component parameter override. A component parameter, defined in a component override area, that updates or replaces a component parameter defined in a component area of the product parameter file. Synonymous with *component override* and *override*.

concurrently. Concerning a mode of operation that includes doing work on two or more activities within a given (short) interval of time.

console. A device used for communications between the operator or maintenance engineer and the computer.

console spooling. Synonym for *virtual console spooling*.

console stack. Refers collectively to the program stack and the terminal input buffer.

control file. In service, a file with file type CNTRL that contains records that identify the updates to be applied and the macro libraries, if any, needed to assemble that source program. A CMS file that is interpreted and directs the flow of a certain process through specific steps. For example, the control file could contain installation steps, default addresses, and PTF prerequisite lists and many other necessary items.

control program. A computer program that schedules and supervises the program execution in a computer system. See *Control Program (CP).*

Control Program (CP). A component of VM/ESA that manages the resources of a single computer so multiple computing systems appear to exist. Each of these apparent systems, or virtual machines, is the functional equivalent of an IBM System/370™, 370-XA, or ESA computer. Also, XC virtual machines provide functions beyond the ESA architecture. See also *virtual machine*.

control section (CSECT). The part of a program specified by the programmer to be a relocatable unit, all elements of which are loaded into adjoining main storage.

control statement. A statement that controls or affects program execution in a data processing system.

copy file. A file having file type COPY that contains nonexecutable real storage definitions that are referred to by macros and assemble files.

copy function. The function initiated by a PF key to copy the contents of a display screen onto an associated hardcopy printer. A remote display terminal copies the entire contents of the screen onto a printer attached to the same control unit. A local display terminal copies all information from the screen, except the screen status information, onto any printer attached to any local display control unit.

COR. Corrective service tape.

corequisite. Corequisites identify other PTFs that must be applied at the same time this PTF is applied. No specific order is required for applying corequisite PTFs.

corequisite change. A change that must be applied to the user's product along with another change. For example, if the user needs to apply change1 to the system and change1 has a corequisite of change2, then the user must apply both change1 and change2 to the system, but not in a specific order. A corequisite change corrects a problem that requires changes to one or more elements of a product or component.

corrective service. Service that IBM supplies on tape to correct a specific problem.

corrective service tape. A tape, supplied by IBM at the user's request, containing a fix for a specific problem and any requisites for the fix.

count-key-data (CKD) device. A DASD that stores data in the format: count field, usually followed by a key field, followed by the actual data of a record. The count field contains the cylinder number, head number, record number, and the length of the data. The key field contains the record's key (search argument).

CP. Control Program.

CP command. A command available to all VM users. Class G CP commands let the general user reconfigure their virtual machine, control devices attached to their virtual machine, do input and output spooling functions, and simulate many other functions of a real computer console. Other CP commands let system operators, system programmers, system analysts, and service representatives manage the resources of the system.

CP directory. Synonym for *VM directory*.

CP read. The condition when CP is waiting for a response or request for work from the user. On a typewriter terminal, the keyboard is unlocked; on a display terminal, the screen status area indicates CP READ.

cross system extensions (CSE). An environment in which end users attached to a single system can participate with additional systems as though all participating systems were one complex.

CSE. Cross system extensions.

CSECT. Control section.

CSL. Callable services library.

cylinder. In a disk pack, the set of all tracks with the same nominal distance from the axis about which the disk pack rotates.

D

DASD. Direct access storage device.

DASD Dump Restore (DDR) program. A service program that copies all or part of a minidisk onto tape, loads the contents of a tape onto a minidisk, or sends data from a DASD or from tape to the virtual printer.

DASD space. Area allocated to DASD units on CKD devices. Area allocated to DASD units on FB-512 devices. Note that *DASD space* is synonymous with *cylinder* when there is no need to differentiate between CKD devices and FB-512 devices.

DBCS. Double-byte character set.

DCSS. Discontiguous saved segment.

DDR program. DASD Dump Restore program.

DELETE. This status, shown in the service-level build status table, indicates the object has been removed from the build list, and the corresponding object must be deleted.

DELETED. This status, listed in the apply status table, indicates that a product has been deleted from the system. In the service-level build status table, it indicates that an object has been deleted from the product.

delimiter. A flag that separates and organizes items of data. Synonymous with *separator*. A character that groups or separates words or values in a line of input. Usually one or more blank characters separate the command name and each operand or option in the command line. In certain cases, a tab, left parenthesis, or backspace character can also act as a delimiter.

Delta disk. In VMSES/E, a minidisk or SFS directory containing a list of the files on a PTF. See program temporary fix (PTF).

Delta string. In VMSES/E, the set of Delta disks.

dependent PTF. A PTF that has another PTF as a prerequisite or corequisite.

dependent requisite. A dependent requisite is a product that must be installed before another product can be installed correctly. Unlike pre-requisites, dependent requisites are no longer satisfied when the requisite product is superseded. This occurs when a product requires a specific level of another product and newer levels of the product will not meet the requirements.

description table. The Software Inventory table that contains the descriptive name for a product, in the system level, and APARs in the service level. The file type of the system level inventory table is SYSDESCT and the file type of the service level inventory table is SRVDESCT.

device support facilities. A program for doing operations on disk volumes so that they can be accessed by IBM and user programs. Examples of these operations are initializing a disk volume and assigning an alternate track.

DIRCONTROL directory. Synonym for *directory control directory*.

direct access storage device (DASD). A storage device in which the access time is effectively independent of the location of the data.

directory. See *auxiliary directory, CMS minidisk file directory, DIRCONTROL directory, directory control directory, file control directory, FILECONTROL directory, SFS directory,* or *VM directory*.

directory identifier (dirid). A fully-qualified directory name (in which the file pool ID and user ID can be allowed to default), a file mode letter, or plus (+) or minus (-) file mode syntax (used in commands).

directory name (dirname). A fully-qualified directory name that can incorporate a period (.) to indicate the user's own top directory (used in commands).

dirid. Directory identifier.

dirname. Directory name.

discontiguous saved segment. One or more segments of storage that were previously loaded, saved, and assigned a unique name. In VM/ESA, a segment begins and ends on a 1MB boundary. The segment can be shared among virtual machines if the segment contains reentrant code. Discontiguous segments used with CMS must be loaded into storage at locations above the address space of a user's CMS virtual machine. They can be detached when no longer needed.

disk. A magnetic disk unit in the user's CMS virtual machine configuration. See *virtual disk*.

display device. An I/O device that gives a visual representation of data.

display terminal. A terminal with a component that can display information on a viewing surface such as a screen or gas panel.

distributed function terminal (DFT). An operational mode that allows multiple concurrent logical terminal sessions. Contrast with *control unit terminal (CUT).*

DMSPARMS file. A CMS file with a file type of DMSPARMS that contains the start-up parameters that SFS file pool server and CRR recovery server processing uses.

double-byte character set (DBCS). A character set that requires 2 bytes to uniquely define each character. This contrasts with EBCDIC, in which each printed character is represented by 1 byte.

dump. To write the contents of part or all of main storage, or part or all of a minidisk, to auxiliary storage or a printer. See *abend dump*.

E

ECKD™. Extended count-key data.

edit. A function that makes changes, additions, or deletions to a file on a disk. These changes are interactively made. The edit function also generates information in a file that did not previously exist.

ERROR. This qualifier of the status field in the service-level build status table indicates that an error was encountered when building an object. In the system-level build status table, it indicates that an error was detected when building a product or object.

ESA virtual machine. A virtual machine that simulates ESA/370 or ESA/390™ functions. Contrast with *370 virtual machine*, *XA virtual machine*, and *XC virtual machine*

exclude list. A file listing PTFs to be omitted from a product or component.

EXEC procedure. A procedure defined by a frequently used sequence of CMS and CP commands to do a commonly required function. A user creates the procedure to save repetitious reentering of the sequence, and invokes the entire procedure by entering a command (that is, the exec file's file name). The

procedure could consist of a long sequence of CMS and CP commands, along with REXX, EXEC 2, or CMS EXEC control statements to control processing within the procedure. A CMS file with a file type of EXEC.

EXEC 2 language. A general-purpose, high-level programming language, particularly suitable for EXEC procedures and XEDIT macros. The EXEC 2 processor runs procedures and XEDIT macros (programs) written in this language. Contrast with *CMS EXEC language* and *Restructured Extended Executor (REXX) language*.

exit. See *user exit* and *installation-wide exit*.

F

FB-512. An FBA device that stores data in 512-byte blocks (refers to DASD devices such as the IBM 9335, 9332, 9313, 3370, and 3310).

FBA. Fixed-block architecture.

feature. A feature is associated with the software distribution order number which has a type, model, and feature field. The feature field identifies a particular deliverable for the given product offering.

file access mode. A file mode number that designates whether the file can be used as a read-only or read/write file by a user. See *file mode*.

file ID. A CMS file identifier that consists of a file name, file type, file mode, or directory ID. The file ID is associated with a particular file when the file is created, defined, or renamed under CMS. See *file name, file type,* and *file mode*.

file mode. A two-character CMS file identifier field containing the file mode letter (A through Z) followed by the file mode number (0 through 6). The file mode letter indicates the minidisk or SFS directory on which the file resides. The file mode number indicates the access mode of the file. See *file access mode*.

file name. A one-to-eight character alphanumeric field, containing A through Z, 0 through 9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

file pool. A collection of minidisks managed by SFS. It contains user files and directories and associated control information. Many users' files and directories can be contained in a single file pool.

file type. A one-to-eight character alphanumeric field, containing A through Z, 0 through 9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is used as a descriptor or as a qualifier of the file name field in the CMS file identifier. See *reserved file types*.

file type abbreviation (ftabbrev). The 3-character PTF abbreviation or the real CMS file type for a part that is not serviced by replacement.

file type abbreviation table. The Software Inventory table that identifies the mapping between PTF-numbered file types and the real CMS file type. The service level inventory does not contain this table.

fixed-block architecture (FBA) device. A disk storage device that stores data in blocks of fixed size or records; these blocks are addressed by block number relative to the beginning of the particular file.

flat file. A file that consists of a set of records ordered by record number or as sequentially entered in the file; a two dimensional file.

free storage. Storage not allocated. The blocks of central storage available for temporary use by programs or by the system.

ftabbrev. File type abbreviation

full-pack minidisk. A virtual disk that contains all of the addressable cylinders of a real DASD volume.

G

GCS. Group Control System for ESA/370 or ESA/390 architecture.

group. Synonym for *virtual machine group*.

Group Control System (GCS). A component of VM/ESA, consisting of a shared segment that the user can IPL and run in a virtual machine. It provides simulated MVS services and unique supervisor services to help support a native SNA network.

GROUP EXEC. A GCS installation tool that prompts you for the specifications needed to build a GCS configuration file.

guest. An operating system running in a virtual machine managed by a VM control program. Contrast with *host*.

H

hard requisites. The hard requisites of a PTF are a subset of its prerequisites. There are two reasons for a prerequisite to be classified as a hard requisite. First, if the PTF depends on a functional change introduced by the requisite, the requisite is considered a hard requisite. For example, the requisite introduces a new

flag and the PTF exploits it. Second, if any of the updates in the PTF affect the same lines of code as the requisite, such that the new update will not apply without the older one, then it is a hard requisite relationship. (Corequisites and if-requisites are by definition hard requisites and are not explicitly listed as hard requisites).

history files. One or more CMS files that describe the changes (with a date and time stamp) made to the VM/ESA system and its installed software products.

I

I/O. Input/output.

if-requisite. At the system-level, an if-requisite lists two products. The first one becomes a requisite product if and only if the second one is installed. At the service-level, an if-requisite lists a PTF in another product that must be applied if and only if the other product is installed.

image library. A set of modules that define the spacing, characters, and copy modification data that a 3800 printer uses to print a spool file or that define the spacing and character set that an impact printer uses to print a spool file. See *system data file*.

initial installation system. In VMSES/E, a functional subset of the VM/ESA system shipped on the VM/ESA system DDR tapes or CD-ROM and used during installation of VM/ESA.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation. A VM user must IPL the specific operating system into the virtual machine that will control the user's work. Each virtual machine can be loaded with a different operating system.

initialize. To set counters, switches, addresses, or contents of storage to starting values.

input/output (I/O). A device whose parts can do an input process and an output process at the same time. A functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

installation-wide exit. An interface to VM/ESA that a system programmer can use to enhance or extend the functions of a VM/ESA system. Generally, an installation-wide exit is activated for all users on the system and is run as part of a system program.

install-time requisites. Product(s) that must be installed before this product can be installed correctly.

interactive. The classification given to a virtual machine depending on this virtual machine's processing characteristics. When a virtual machine uses less than its allocation time slice because of terminal I/O, the virtual machine is classified as being interactive. Contrast with *noninteractive*.

interface. A shared boundary between two or more entities. An interface might be a hardware or software component that links two devices or programs together.

interrupt. A suspension of a process, such as execution of a computer program, caused by an external event and done in such a way that the process can be resumed.

invoke. To start a command, procedure, or program.

IPL. Initial program load.

L

line number. A number located at either the beginning or the end of a record (line) that can be used during editing to refer to that line. See *prompting*.

load. In installation and service, to move files from tape to disk, auxiliary storage to main storage, or minidisks to virtual storage within a virtual machine.

load map. A map containing the storage addresses of control sections and entry points of a program loaded into storage.

loadable unit. A portion of a product that can be installed independently of the rest of the product, but is serviced as part of the product.

loader. A routine, commonly a computer program, that reads data into main storage.

Local disk. In VMSES/E, a minidisk or SFS directory containing local modifications, customized files, and any circumventive service.

local modification. Any change applied to a product other than a PTF See circumventive service and user modification.

local service. Changes manually applied to a product or component (that is, not using the program update service or corrective service procedures). See *circumventive service* and *user modification*.

local tracking number. The unique identifier assigned to a local modification. The local tracking number is used in the file type of update files and in the update file identification records of auxiliary control files. Each installation has its own system of local tracking numbers.

Local string. In VMSES/E, the set of Local disks.

logical record. A formatted record that consists of a 2-byte logical record length and a data field of variable length.

logical saved segment. A portion of a physical saved segment that CMS can manipulate. Each logical segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language repositories, user-defined objects, or a single minidisk directory. A system segment identification file (SYSTEM SEGID) associates a logical saved segment to the physical saved segment in which it resides. See *physical saved segment* and *saved segment*.

logoff. The procedure by which a user ends a terminal session.

logon. The procedure by which a user begins a terminal session.

low common storage. GCS common storage that resides below the 16MB line. See *common storage*.

M

machine. A synonym for a virtual machine running under the control of VM/ESA.

macro. Synonym for *macrodefinition* and *macroinstruction*.

macro library. A library of macrodefinitions.

macrodefinition. 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. Synonymous with *macro*.

macroinstruction. In assembler language programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macrodefinition. The statements usually produced from the macrodefinition replace the macroinstruction in the program. Synonymous with *macro*.

MANUAL. This status, listed in the service-level build status table, indicates that the object requires MANUAL processing.

map. In CMS, the file that contains a CMS output listing, such as (1) a list of macros in the MACLIB library, including macro size and location within the library; (2) a listing of the directory entries for the DOS/VS system or private source, relocatable, or core image libraries; (3) a linkage editor map for CMS/DOS programs; and (4) a module map containing entry point locations.

mapping. To show relationships between objects.

MB. Megabyte.

MDISK. Another name for minidisk. The VM directory statement that describes a user's storage space.

megabyte (MB). 1,048,576 bytes.

member saved segment. A saved segment that begins and ends on a page boundary. It can be a member in up to 64 segment spaces and is accessed either by the segment space name or by its own name. Contrast with *discontiguous saved segment*. See *saved segment*, *segment*, and *segment space*.

memo-to-users. A file provided on a service tape that contains specific service information for a product.

merge. When receiving files from a service tape using VMFMRDSK, the process of moving existing service files from each minidisk or SFS directory in the target string to the minidisk or directory that contains the previous service level. The result is that the primary target minidisk or directory is left empty and ready to receive the latest service.

message. Data sent from a source application to a target application program in a conversation. See also *message text*, *message key*, and *message header*.

minidisk. A logical subdivision (or all) of a physical disk pack that has its own virtual device address, consecutive virtual cylinders (starting with virtual cylinder 0), and a VTOC or disk label identifier. Each user virtual disk is preallocated and defined by a VM/ESA directory entry as belonging to a user.

minidisk directory. Synonym for *CMS minidisk file directory*.

module. A unit of a software product that is discretely and separately identifiable with respect to modifying, compiling, and merging with other units, or with respect to loading and execution. For example, the input to, or output from, a compiler, the assembler, the linkage editor, or an exec routine. A nonrelocatable file whose external references have been resolved.

N

named saved system (NSS). A copy of an operating system that a user has named and saved in a file. The user can load the operating system by its name, which is more efficient than loading it by device number. See *discontiguous saved segment*, *member saved segment*, *saved segment*, *segment space*, and *system data file*.

negative prerequisite. In VMSES/E, a product that cannot exist on a system at the same time as another product.

NSS. Named saved system.

nucleus. The part of CP and CMS resident in main storage.

O

object. In VMSES/E, a usable form defined in build lists. A built part of a product. A product consists of many objects, for example, nuclei, modules, execs, help files, and macro libraries. See usable forms.

Compare *subject*.

object code. Compiler or assembler output that is executable machine code or is suitable for more processing to produce executable machine code. Contrast with *source code*.

object module. A module that is the output of an assembler or a compiler and is input to a linkage editor.

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.

out-of-component requisite. In VMSES/E, at the service-level, a PTF from another product that must be applied to that product in order for this PTF to function properly.

overhead. The additional processor time charged to each virtual machine for the CP functions needed to simulate the virtual machine environment and for paging and scheduling time.

override. Synonym for *component parameter override*.

override area. Synonym for *component override area*.

override file. Synonym for *class override file* and *product parameter override file*.

override $PPF. Synonym for *override product parameter file*.

P

pack. A set of flat, circular recording surfaces that a disk storage device uses. A disk pack.

page. A fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage.

parameter. A variable that is given a constant value for a specified application and that may denote the application.

parameter driven installation (PDI). A product format that lets you specify a product installation location, specify installation related parameters, install multiple copies of a product, and select a default installation path.

part. A CMS file provided on a VM/ESA System DDR tape, CD-ROM, or service tape as input to the build process. See *build*. A part is the smallest serviceable unit of a component.

part handler. An exec provided by VMSES/E that builds a specific type of object or loads parts from service media.

parts catalog. In VMSES/E, a set of Software Inventory files that catalog all parts of a product on a minidisk or SFS directory. All product parts are cataloged when they are loaded onto the system, when they are generated, and when they are moved.

password. In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it.

patch. A circumventive service change applied directly to object code in a text deck in a nucleus.

patch update file. A file containing a single patch. The file can also specify requisites for applying the patch.

PDI. Parameter driven installation.

PF key. Programmed function key.

physical saved segment. One or more pages of storage that have been named and retained on a CP-owned volume (DASD). Once created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment to its logical saved segments. See *logical saved segment* and *saved segment*.

PPF. Product parameter file.

preferred auxiliary file. In CMS, an auxiliary file that applies to a particular version of a source module to be updated, if multiple versions of the module exist.

preferred virtual machine. A particular virtual machine that has one or more of the performance options assigned to it.

prefix area. The five left-most positions on the XEDIT full-screen display, in which prefix subcommands or prefix macros can be entered. See *prefix macros* and *prefix subcommands*.

prefix macros. XEDIT macros entered in the prefix area of any line on a full-screen display. See *prefix area*.

prefix subcommands. XEDIT subcommands entered in the prefix area of any line on a full-screen display. See *prefix area*.

prerequisite. In VMSES/E, at the system-level, a product that must be installed before another product can be installed. At the service-level, a PTF that must be installed before another product can be installed.

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 the user must apply change1 before applying change2.

preventive service. The application of all PTFs from a PUT or RSU. Contrast with selective preventive service. See program update tape and product service upgrade.

primary system operator privilege class. The CP privilege class A user. This operator has primary control over the VM/ESA system and can enable and disable teleprocessing lines, lock and unlock pages, force users off the VM/ESA system, issue warning messages, query, and set (and reset) performance options for selected virtual machines, and invoke VM/ESA accounting. If the current primary system operator logs off, the next class A user to log on becomes the primary system operator.

private storage. A combination of application code and GCS code available to only one particular virtual machine. No virtual machine can access or share another's private storage area.

privilege class. One or more classes assigned to a virtual machine user in a VM/ESA directory entry; each privilege class specified lets a user access a logical subset of the CP commands. There are nine IBM-defined privilege classes that correspond to specific administrative functions. They are:

Class A - primary system operator
Class B - system resource operator
Class C - system programmer
Class D - spooling operator
Class E - system analyst
Class F - service representative
Class G - general user
Class H - reserved for IBM use
Class Any - available to any user.

The privilege classes can be changed to meet the needs of an installation. See *class authority* and *user class restructure (UCR)*.

privileged program. In GCS, a program called by a GCS application that operates in supervisor state and uses privileged functions. A privileged program is one that meets either of the following requirements:

- It runs in an authorized virtual machine.
- It is called through the AUTHCALL facility.

Synonymous with *authorized program*. Contrast with *nonprivileged program*.

process. A systematic sequence of operations to produce a specified result. A process is usually logical, not physical. In CMS Multitasking, a collection of threads performing related work. A process can have resources associated with it, such as storage subpools, queues, open files, and APPC conversations. All threads in a process have equal access to the resources associated with the process.

PRODPART file. VMSES/E uses information in this file, included on a product's install tape, to update entries in the system-level Software Inventory each time a product is loaded onto your system.

product. Any separately installable software program, whether supplied by IBM or otherwise, distinct from others and recognizable by a unique identification code. The product identification code is unique to a given product, but does not identify the release level of that product.

product identifier (prodid). The product identifier is the 7- or 8-alphanumeric character identifier assigned to the product by IBM.

product parameter file (PPF). A file containing installation and service parameters for a product: control options, minidisk and SFS directory assignments, and component part type/function lists.

product parameter override file. A file containing one or more component override areas.

product processing exit. An interface used by program products to perform additional product installation tasks.

product service upgrade (PSU). A procedure used to upgrade the service level of a product or component using a recommended service upgrade (RSU) tape.

product tape. One of a set of tapes containing individual components or products to load and build.

PROFILE EXEC. A special EXEC procedure with a file name of PROFILE that a user can create. The

procedure is usually executed immediately after CMS is loaded into a virtual machine (also known as IPL CMS).

program temporary fix (PTF). Code changes needed to correct a problem reported in an APAR. The corrected code is included in later releases. A PTF contains one or more APAR fixes. For object-maintained parts that are changed, the PTF includes replacement parts. For source-maintained parts that are changed, the PTF includes update files and replacement parts. 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.

program update service. Receiving service from a PUT or RSU, applying all or some of the changes, and rebuilding the serviced parts. See preventive service and selective preventive service.

program update tape (PUT). A tape containing a customized collection of service tapes (preventive service) to match the products listed in a customer's ISD (IBM Software Distribution) profile. Each PUT contains cumulative service for the customer's products back to earlier release levels of the product still supported. The tape is distributed to authorized customers of the products at scheduled intervals or on request.

programmed function (PF) key. On a terminal, a key that can do various functions selected by the user or determined by an application program.

prompt. A displayed message that describes required input or gives operational information.

prompting. An interactive technique that lets the program guide the user in supplying information to a program. The program types or displays a request, question, message, or number, and the user enters the desired response. The process is repeated until all the necessary information is supplied.

PSU. product service upgrade

PTF. Program temporary fix.

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 for different products or different releases of a product have different numbers.

PUT. Program update tape.

R

R/O. Read-only.

R/W. Read/write.

rdev. The real device address of an I/O device.

reach-ahead service. Corrective service or local service that has been applied to a product but is not available on a program update tape, product service upgrade, or other service vehicle.

read authority. The authority to read the contents of a file without being able to change them. For a directory, read authority lets the user view the names of the objects in the directory.

read-only access. An access mode associated with a virtual disk or SFS directory that lets a user read, but not write or update, any file on the disk or SFS directory.

read/write access. An access mode associated with a virtual disk or SFS directory that lets a user read and write any file on the disk or SFS directory (if write authorized).

real address. The address of a location in real storage or the address of a real I/O device.

receive. Bringing into the specified buffer data sent to the user's virtual machine from another virtual machine or from the user's own virtual machine. To load service files from a service tape. In CMS Multitasking interprocess communication, the action of retrieving a message from a queue.

receive ID. A 7- or 8- alphanumeric character identifier that is used to name the Software Inventory files created during receive processing.

receive status table. The Software Inventory table that contains the relationship between a product and the $PPF file used to install it. It also identifies what products of PTFs have been received or committed. The file type of the system level inventory table is SYSRECS and the file type of the service level inventory table is SRVRECS.

RECEIVED. This status, listed in the receive status table, indicates that a product or PTF has been RECEIVED on the system.

Recommended Service Upgrade (RSU) tape. A tape containing preventive service for upgrading the current release of a VM/ESA system once it has been installed.

recomp. To change the number of cylinders or blocks on the disk that are available to you.

regression. Causing serviced parts to go back to earlier levels. This can occur when applying changes from a PUT to parts updated by corrective service or user modifications.

Remote Spooling Communications Subsystem Networking (RSCS). An IBM licensed program and special-purpose subsystem that supports the reception and transmission of messages, files, commands, and jobs over a computer network.

REMOVED. This status, listed in the apply status table, indicates that the PTF has been REMOVED from the system.

replacement parts. See serviceable parts.

replacement service. Servicing a part by replacing the part with a new one.

requisite. The requirements of a product or PTF.

requisite relationships. The interrelated requirements of a product or PTF.

requisite table. The Software Inventory table that contains the requisite relationships between products, in the system level, and PTFs in the service level. The file type of the system level inventory table is SYSREQT and the file type of the service level inventory table is SRVREQT.

resource. A program, a data file, a specific set of files, a device, or any other entity or a set of entities that the user can uniquely identify for application program processing in a VM system.

REXX exec. An EXEC procedure or XEDIT macro written in the REXX language and processed by the REXX/VM Interpreter. Synonymous with *REXX program*.

REXX program. Synonym for *REXX exec*.

RSCS. Remote Spooling Communications Subsystem Networking.

RSU. Recommended Service Upgrade

S

saved segment. A segment of storage that has been saved and assigned a name. The saved segments can be physical saved segments that CP recognizes or logical saved segments that CMS recognizes. The segments can be loaded and shared among virtual machines, which helps use real storage more efficiently, or a private, nonshared copy can be loaded into a virtual machine. See *logical saved segment* and *physical saved segment*.

saved system. A special nonrelocatable copy of a virtual machine's virtual storage and associated registers kept on a CP-owned disk and loaded by name instead of by I/O device address. Loading a saved system by name substantially reduces the time it takes to IPL the system in a virtual machine. Also, a saved system such as CMS can also share one or more 1MB segments of reenterable code in real storage between virtual machines. This reduces the cumulative real main storage requirements and paging demands of such virtual machines.

screen. An illuminated display surface; for example, the display surface of a CRT. Synonymous with *physical screen*.

SDO. System delivery offering.

secondary user. When a user is disconnected — that is, has no virtual console on line — a secondary user can be designated to receive the disconnected user's console messages and to enter commands to the disconnected user's console.

segment. In System/370 architecture, 64KB of virtual storage. In 370-XA, ESA/370, ESA/390, and ESA/XC architecture, 1MB of virtual storage.

segment space. A saved segment is composed of up to 64 member saved segments accessed by a single name. A segment space occupies one or more architecturally-defined segments. It begins and ends on a 1MB boundary. A user with access to a segment space has access to all of its members. See *discontiguous saved segment*, *member saved segment*, *saved segment*, and *segment*.

select data file. In VMSES/E, a file containing a list of the parts serviced by the VMFAPPLY EXEC. The VMFAPPLY EXEC updates this file with a time stamp and a list of parts that were serviced. The VMFBLD EXEC checks the select data file for build requirements and updates the objects that are affected by service to a status of 'SERVICED' in the service-level build status table. The select data file is named appid $SELECT, where appid is the apply ID. See apply ID.

selective preventive service. The selective application of PTFs from a PUT or RSU. Contrast with preventive service.

separator. Synonym for *delimiter*.

server. The general name for a virtual machine that provides a service for a requesting virtual machine.

service. Changing a product after installation. See *corrective service, local service,* and *program update service*.

service level. The PTF and preventive service level that is associated with the testing level and support level of an orderable product function.

service level inventory. see *service-level Software Inventory*.

service-level Software Inventory. In VMSES/E, the level of the Software Inventories that contains: requisite relationships between PTFs, the status of PTFs installed, the service level of each part of the product and, the status of objects built for the product.

service machine. A virtual machine running a program that provides system-wide services.

service tape. A tape containing service changes for one or more products. See *corrective service tape* and *recommended service upgrade (RSU)*.

service virtual machine. A virtual machine that provides a system service such as accounting, error recording, monitoring, or that provided by a supported licensed program.

serviceable parts. The individual parts of a product that can be serviced separately. A serviceable part has the file name of the source or replacement part and a file type in the form tttnnnnn, where ttt is a unique three-character abbreviation for the part type and nnnnn is the PTF number. Serviceable parts are maintained by both source updates and replacement service.

SERVICED. This status, listed in the service-level build status table, indicates that the object has been SERVICED but not built.

SFS. Shared file system.

SFS directory. A group of files. SFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files.

Shared File System (SFS). A part of CMS that lets users organize their files into groups known as *directories* and selectively share those files and directories with other users.

shared segment. A feature of a saved system or physical saved segment that lets one or more segments of reentrant code or data in real storage be shared among many virtual machines. For example, if a saved CMS system was generated, the CMS nucleus is shared in real storage among all CMS virtual machines loaded by name; that is, every CMS machine's segment of virtual storage maps to the same 1MB of real storage. See *discontiguous saved segment* and *saved system*.

shared system. See *saved system* and *shared read-only system residence disk*.

simultaneous peripheral operations online (SPOOL). (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

single user group. The concept in GCS of a virtual machine that runs applications that do not require group communications. This allows an application to run without the overhead of group initialization and multiple virtual machines. Multiple users can IPL the same saved system if it had been built for a single user environment. See *virtual machine group*.

SNA. Systems Network Architecture.

soft requisite. The subset of a PTF's requisite that is not a hard requisite. A PTF has a soft requisite for another PTF if it affects any of the same modules. The relationship exists because the pre-built replacement parts that are shipped with PTFs are built with all prior PTFs.

software inventory management. Utilities provided by VMSES/E that provide a standard interface to the system level inventories, service level inventories, tool control statements (TCS), product parameter file (PPF), and file type abbreviation table.

software product. Any software supplied by IBM or an Original Equipment Manufacturer (OEM), or user written programs. The term includes program offerings and program products (PPs).

source code. The input to a compiler or assembler, written in a source language. Contrast with *object code*.

source file. A file that contains source statements for such items as high-level language programs and data description specifications.

source product parameter file. In VMSES/E, a file supplied with a product containing: recommended values for the options that control VMSES/E processing for the product, installation and service tape formats, and the list of build lists used to build the product. The file name of the source product parameter file matches the prodid of the product and the file type is $PPF. Source PPFs.

source update. A change to the original assembler code provided with a product. VM source code is contained in files with a file type of ASSEMBLE. To update an ASSEMBLE file, the user creates update files containing control statements that describe the changes to be made.

source update file. A file containing a single change to a statement in a source file. The file can also include requisite information for applying the change. Synonymous with *update file*.

SPOOL. Simultaneous peripheral operations online.

spool file. A collection of data along with CCWs for processing on a unit record device. Contrast with *system data file*.

spool ID. A spool file identification number automatically assigned by CP when the file is closed. The spool ID number can be from 0001 to 9900; it is unique for each spool file. To identify a given spool file, a user must specify the owner's user ID, the virtual device type, and the spool ID.

spooling. The processing of files created by or intended for virtual readers, punches, and printers. The spool files can be sent from one virtual device to another, from one virtual machine to another, and to real devices. See *virtual console spooling*.

stand-alone dump. A dump acquired without regular system functions. For example, to obtain a CP dump when the regular system is unable to dump the machine, the stand-alone dump facility gets a CP stand-alone dump.

string. A group of minidisks defined for a specific function in the product parameter file, for example, the BASE2 string, which holds source code.

sub hard requisite. In VMSES/E, a sub hard requisite is a hard requisite of an explicitly defined requisite.

sub if-requisite. In VMSES/E, a sub if-requisite is an if-requisite of an explicitly defined requisite.

subcommand. The commands of processors such as EDIT or XEDIT that run under CMS.

subdirectory. Any SFS directory below a user's top directory. The CREATE DIRECTORY command creates subdirectories. There can be up to eight levels of subdirectories with no limit on the number of them at each level, other than overall DASD space limits. Each level of a subdirectory is an additional identifier of up to 16 characters that is appended to next higher level subdirectory.

subrequisite. A subrequisite is a prerequisite or corequisite or an explicitly defined requisite. The requisite of requisites.

SUPED. This status, listed in the service-level apply status table, indicates that the PTF has been superseded.

supersede. When a PTF supersedes another PTF, it includes all of the APARs, parts and requisite relationships of the PTF it supersedes.

syntax. The rules for the construction of a command or program.

system administrator. The person responsible for maintaining a computer system.

system DDR tape. A tape containing the image of a built system for each type of DASD.

system delivery offering (SDO). A VM/ESA package that includes a subset of all VM products or components. This package has a single point of order and delivery, is refreshed periodically, and is installed from one logical tape. All products or components included with the package, and their requisite relationships, are tested to ensure the package functions as a system.

System disk. In VMSES/E, a minidisk or SFS directory containing other products that are required during service.

system level inventory. See *system-level Software Inventory*.

system-level Software Inventory. Level of the Software Inventories that contains: requisite relationships between products or components, the status of the product or component on the system, mapping of product identifier to the name of the product parameter file used during installation, and mapping of PTF file type abbreviation to real CMS file type.

system offering. A package containing VM/SP and associated products.

system profile. An EXEC (SYSPROF) that resides in a saved system or on a system disk and called by CMS initialization. It contains some initialization functions, and provides a means for installations to override the default CMS environment by tailoring the exec to suit the installation.

system restart. The restart that allows reuse of previously initialized areas. System restart usually requires less time than IPL. See *warm start*.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

System string. In VMSES/E, the set of System disks.

T

T-disk. Synonym for *temporary disk*.

tailorable file. any source level product file that requires user input in order for the product to work correctly. (An example is a PROFILE EXEC.)

tailorings. Changes made to a source level product file to customize it for your own environment.

tape descriptor file. A file containing a directory of the products on a service tape.

tape document. A document describing the service procedure for a service tape.

target. One of many ways to identify a line to be searched for by XEDIT. A target can be specified as an absolute line number, a relative displacement from the current line, a line name, or a string expression.

Target disk. In VMESE/E, a minidisk of SFS directory to which tape files are received on which the objects are built.

Target string. In VMSES/E, the set of Target disks.

task. A basic unit of work used for the execution of a program or a system function.

temporary disk. An area on a DASD available to the user for newly created or stored files until logoff, at which time the area is released. Temporary disk space is allocated to the user during logon or when entering the CP DEFINE command. Synonymous with *T-disk*.

temporary product parameter file. In VMSES/E, the output of the VMFOVER EXEC. The file name is either the file name of the last override product parameter file in the chain of overrides, or the file name of the source product parameter file. The file type is $PPFTEMP.

terminal. A device, usually equipped with a keyboard and a display, capable of sending and receiving information.

text deck. An object-code file that must be additionally processed to produce executable machine code.

text library. A CMS file that contains relocatable object modules and a directory that indicates the location of each of these modules within the library.

time stamp. A record containing the TOD clock value stored in its internal 32-bit binary format.

time-of-day (TOD) clock. A hardware feature required by VM/ESA. The TOD clock is incremented once every microsecond, and provides a consistent measure of elapsed time suitable for the indication of date and time; it runs regardless of the processor state (running, wait, or stopped).

TOD clock. Time-of-day clock.

token. An eight-character symbol created by the CMS EXEC processor when it scans an EXEC procedure or EDIT macro statements. Symbols longer than eight characters are truncated to eight characters.

U

update file. Synonym for *source update file*.

update service. Servicing a part by applying a change to a source file statement, then assembling or compiling the source file to produce a new object file.

usable form. In VMSES/E, a part of a product whose level cannot be identified from its file name or file type, the final objects which make up the product, a source file with the file type ASSEMBLE. If the level can be identified from its file name the part is referred to as a serviceable part.

usable form product parameter file. Product parameter files used by the majority of VMSES/E execs. The file name matches the file name of either the last override product parameter file in the chain of overrides, or the file name of the source product parameter file if there are no overrides. The file type is PPF.

user. Anyone who requests the services of a computing system.

user class. A privilege category assigned to a virtual machine user in the user's directory entry; each class specified allows access to a logical subset of all the CP commands. See *privilege class*.

user exit. An interface to VM/ESA that can be used by an application program. Generally, a user exit affects only the particular application specifying the exit and is run as part of the application program.

user ID. User identification.

user memo. At the system-level, special instructions for installing a product, and at the service-level, special instructions for installing a PTF.

user modification. Any change that a user originates for a product or component.

V

vaddr. Virtual address.

variable symbol. In an EXEC procedure, a symbol beginning with an ampersand (&) character, the value of which is assigned by the user, or sometimes by the VM/REXX interpreter, the EXEC 2 processor, or CMS EXEC processor. The value of a variable symbol can be tested and changed using control statements. See *special variable*.

version vector table. The Software Inventory table that identifies which PTFs have been applied to each part of the product and the current level of each part. The file type of the service level inventory table is VVT*lvlid*. The *lvlid* may be unique for each level of service the customer has installed for a product or component. It corresponds directly to each AUX level in the control file. The system level inventory does not contain this table.

virtual address. The address of a location in virtual storage. A virtual address must be translated into a real address to process the data in processor storage.

virtual console. A console simulated by CP on a terminal such as a 3270. The virtual device type and I/O address are defined in the VM/ESA directory entry for that virtual machine.

virtual console spooling. The writing of console I/O on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at any time. When the console spool file is closed, it becomes a printer spool file. Synonymous with *console spooling*.

virtual disk. A logical subdivision (or all) of a physical disk storage device that has its own address, consecutive storage space for data, and an index or description of the stored data so that the data can be accessed. A virtual disk is also called a minidisk. See *disk*.

virtual machine (VM). A functional equivalent of a computing system. In VM/ESA, virtual machines can simulate the System/370, 370-XA, ESA/370, and ESA/390 functions. In addition, on ESA/390 systems, the XC virtual machine architecture is available. Each virtual machine is controlled by an operating system. VM controls the concurrent execution of several virtual machines on an actual processor complex. See *370 virtual machine*, *XA virtual machine*, *ESA virtual machine*, and *XC virtual machine*.

virtual machine group. The concept in GCS of two or more virtual machines associated with each other through the same named system (for example, IPL GCS1). Virtual machines in a group share common read/write storage and can communicate with one another through facilities provided by GCS. Synonymous with *group*. See *single user group*.

Virtual Machine/Enterprise Systems Architecture (VM/ESA). IBM trademarks for a licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a *real* machine.

virtual printer (or punch). A printer (or card punch) simulated on disk by CP for a virtual machine. The virtual device type and I/O address are usually defined in the VM/ESA directory entry for that virtual machine.

virtual storage. Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, not by the actual number of main storage locations.

virtual=real area (V=R area). The part of real storage, starting with real page 1, where a virtual=real machine can execute. CP maintains control of real page zero; only page zero of the virtual=real machine is relocated. Only one virtual machine at a time can occupy the virtual=real area. The area must be defined during VM/ESA system generation to contain the largest virtual=real machine likely to run. See *virtual=real option*.

VM. Virtual machine.

VM directory. A CP disk file that defines each virtual machine's typical configuration: the user ID, password, regular and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired. Synonymous with *CP directory*.

VM/ESA. See *Virtual Machine/Enterprise Systems Architecture.*

VMLIB. The name of the CSL supplied with VM/ESA and that contains routines to do various VM functions.

VMSES. A component of VM in VM/ESA Rel. 1.0 that provides the tools for installing and servicing the various components of the VM product.

VMSES/E. Virtual Machine Serviceability Enhancements Staged/Extended.

VMSES/E. A component of VM, first shipped in VM/ESA Rel. 1.1, that provides the tools for installing and servicing the various components of the VM product. It is also the strategic installation and service tool for all of the other products that run on VM/ESA platforms.

VMSES/E installation/service tool. Consists of two VMSES/E user interfaces, VMFINS and VMFSIM, all of the VMSES/E commands, and the service-level and system-level Software Inventories. Synonymous with *VMSES/E*.

volid. Volume identifier.

volume identifier (volid). The volume identification label for a disk.

W

warm start. The result of an IPL that does not erase previous system data. The automatic reinitialization of the VM/ESA control program that occurs if the control program cannot continue processing. Closed spool files and the VM/ESA accounting information are not lost. Contrast with *checkpoint (CKPT) start, cold start*, and *force start*.

window. An area on the physical screen where virtual screen data can be displayed. Windowing lets the user do such functions as defining, positioning, and overlaying windows; scrolling backward and forward through data; and writing data into virtual screens.

write authority. The authority to read or change the contents of a file or directory. Write authority implies read authority.

X

XA mode. A GCS mode of operation on ESA that uses the full capabilities of the Extended Systems Architecture.

XEDIT. The CMS facility, containing the XEDIT command and XEDIT subcommands and macros, that lets a user create, change, and manipulate CMS files.

XEDIT macro. A procedure defined by a frequently used command sequence to do a commonly required editing function. A user creates the macro to save repetitious reentering of the sequence, and invokes the entire procedure by entering a command (that is, the macro file's file name). The procedure can consist of a long sequence of XEDIT commands and subcommands or both, and CMS and CP commands or both, along with REXX or EXEC 2 control statements to control processing within the procedure. A CMS file with a file type of *XEDIT*.

Y

Y-STAT. A block of storage that contains the FSTs associated with file mode Y. The FSTs are sorted so that a binary search can search for files. The Y-STAT usually resides in the CMS nucleus so it can be shared. Only files with file mode of 2 will have their associated FSTs in the Y-STAT.

Z

zap. To modify or dump an individual text file, using the ZAP command or the ZAPTEXT EXEC.

3

3262. Refers to the IBM 3262 Printer, Models 1 and 11.

3270. Refers to a series of IBM display devices, for example, the IBM 3275, 3276 Controller Display Station; 3277, 3278, and 3279 Display Stations; the 3290 Information Panel; and the 3287 and 3286 printers. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also refers to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

3284. Refers to the IBM 3284 Printer. Information on the 3284 also pertains to the IBM 3286, 3287, 3288, and 3289 printers, unless otherwise noted.

3380. Refers to the IBM 3380 Direct Access Storage Device.

3390. Refers to the IBM 3390 Direct Access Storage Device.

3422. Refers to the IBM 3422 Magnetic Tape Subsystem.

3480. Refers to the IBM 3480 Magnetic Tape Subsystem.

3490. Refers to the IBM 3490 Magnetic Tape Subsystem.

370 mode. A GCS mode of operation on ESA that simulates 370 architecture.

370 virtual machine. A virtual machine that simulates System/370 functions. Contrast with *XA virtual machine, ESA virtual machine,* and *XC virtual machine*.

3800. Refers to the IBM 3800 Printing Subsystems. A specific device type is used only when a distinction is required between device types.

4

4245. Refers to the IBM 4245 Printer.

4248. Refers to the IBM 4248 Printer.

4250. Refers to the IBM 4250 Printer.

9

9332. Refers to the IBM 9332 Direct Access Storage Device, Model 400.

9335. Refers to the IBM 9335 Direct Access Storage Device, Models A01 and B01.

9370. Refers to a series of processors, namely the IBM 9371 Models 10, 12, and 14, the IBM 9373 Model 20, the IBM 9375 Models 40 and 60, the IBM 9377 Model 90, and other models.

Bibliography

This bibliography lists the publications that provide information about your VM/ESA system. The VM/ESA library includes VM/ESA base publications, publications for additional facilities included with VM/ESA, and publications for VM/ESA optional features.

VM/ESA publications may be available as Adobe Portable Document Format (PDF) files, IBM BookManager® files, or printed books. For abstracts of VM/ESA publications and other library-related information, including current editions and available publication formats, see *VM/ESA: General Information*.

VM/ESA Base Publications

Evaluation

VM/ESA: Licensed Program Specifications, GC24-5744

VM/ESA: General Information, GC24-5745

Installation and Service

VM/ESA: Installation Guide, GC24-5836

VM/ESA: Service Guide, GC24-5838

VM/ESA: VMSES/E Introduction and Reference, GC24-5837

Planning and Administration

VM/ESA: Planning and Administration, SC24-5750

VM/ESA: CMS File Pool Planning, Administration, and Operation, SC24-5751

VM/ESA: Conversion Guide and Notebook, GC24-5839

VM/ESA: REXX/EXEC Migration Tool for VM/ESA, GC24-5752

VM/ESA: Running Guest Operating Systems, SC24-5755

VM/ESA: Connectivity Planning, Administration, and Operation, SC24-5756

VM/ESA: Group Control System, SC24-5757

VM/ESA: Performance, SC24-5782

Customization

IBM VM/ESA: CP Exit Customization, SC24-5672

Operation

VM/ESA: System Operation, SC24-5758

VM/ESA: Virtual Machine Operation, SC24-5759

Application Programming

VM/ESA: CP Programming Services, SC24-5760

VM/ESA: CMS Application Development Guide, SC24-5761

VM/ESA: CMS Application Development Reference, SC24-5762

VM/ESA: CMS Application Development Guide for Assembler, SC24-5763

VM/ESA: CMS Application Development Reference for Assembler, SC24-5764

VM/ESA: CMS Application Multitasking, SC24-5766

VM/ESA: REXX/VM Primer, SC24-5598

VM/ESA: REXX/VM User's Guide, SC24-5465

VM/ESA: REXX/VM Reference, SC24-5770

IBM VM/ESA: Distributed Graphical User Interface Toolkit, SC24-5724

IBM VM/ESA: Reusable Server Kernel Programmer's Guide and Reference, SC24-5852

VM/ESA: Enterprise Systems Architecture/Extended Configuration Principles of Operation, SC24-5594

VM/ESA: Programmer's Guide to the Server-Requester Programming Interface for VM, SC24-5455

VM/ESA: CPI Communications User's Guide, SC24-5595

Common Programming Interface Communications Reference, SC26-4399

Common Programming Interface Resource Recovery Reference, SC31-6821

External Security Interface (RACROUTE) Macro Reference for MVS and VM, GC28-1366

End Use

VM/ESA: CP Command and Utility Reference, SC24-5773

VM/ESA: CMS Primer, SC24-5458

VM/ESA: CMS User's Guide, SC24-5775

VM/ESA: CMS Command Reference, SC24-5776

IBM VM/ESA: Graphical User Interface Facility, SC24-5789

VM/ESA: CMS Pipelines User's Guide, SC24-5777

VM/ESA: CMS Pipelines Reference, SC24-5778

CMS/TSO Pipelines: Author's Edition, SL26-0018

VM/ESA: XEDIT User's Guide, SC24-5779

VM/ESA: XEDIT Command and Macro Reference, SC24-5780

VM/ESA: Quick Reference, SX24-5290

Diagnosis

VM/ESA: System Messages and Codes, GC24-5841

VM/ESA: Dump Viewing Facility, GC24-5853

VM/ESA: Diagnosis Guide, GC24-5854

VM/ESA: CP Diagnosis Reference, SC24-5855

VM/ESA: CP Diagnosis Reference Summary, SX24-5292

VM/ESA: CMS Diagnosis Reference, SC24-5857

Note: CP and CMS control block information is not provided in book form. This information is available on the IBM VM/ESA operating system home page (http://www.ibm.com/s390/vm).

Publications for Additional Facilities

OpenEdition® for VM/ESA

IBM OpenEdition for VM/ESA: POSIX Conformance Document, GC24-5842

IBM OpenEdition for VM/ESA: User's Guide, SC24-5727

IBM OpenEdition for VM/ESA: Command Reference, SC24-5728

IBM OpenEdition for VM/ESA: Advanced Application Programming Tools, SC24-5729

IBM OpenEdition for VM/ESA: Callable Services Reference, SC24-5726

IBM OpenEdition for VM/ESA: Sockets Reference, SC24-5741

IBM C for VM/ESA: Library Reference, SC23-3908

Debug Tool User's Guide and Reference, SC09-2137

DFSMS/VM®

VM/ESA: DFSMS/VM Function Level 221 Planning Guide, GC35-0121

VM/ESA: DFSMS/VM Function Level 221 Installation and Customization, SC26-4704

VM/ESA: DFSMS/VM Function Level 221 Storage Administration Guide and Reference, SH35-0111

VM/ESA: DFSMS/VM Function Level 221 Removable Media Services User's Guide and Reference, SC35-0141

VM/ESA: DFSMS/VM Function Level 221 Messages and Codes, SC26-4707

VM/ESA: DFSMS/VM Function Level 221 Diagnosis Guide, LY27-9589

S/390® Open Systems Adapter Support Facility for VM/ESA

Planning for the System/390 Open Systems Adapter Feature, GC23-3870

IBM VM/ESA: Open Systems Adapter Support Facility User's Guide, SC28-1992

Language Environment®

Language Environment for OS/390 & VM: Concepts Guide, GC28-1945

Language Environment for OS/390 & VM: Migration Guide, SC28-1944

Language Environment for OS/390 & VM: Programming Guide, SC28-1939

Language Environment for OS/390 & VM: Programming Reference, SC28-1940

Language Environment for OS/390 & VM: Writing Interlanguage Communication Applications, SC28-1943

Language Environment for OS/390 & VM: Debugging Guide and Run-Time Messages, SC28-1942

Publications for Optional Features

CMS Utilities Feature

VM/ESA: CMS Utilities Feature, SC24-5535

TCP/IP Feature for VM/ESA

VM/ESA: TCP/IP Function Level 320 Planning and Customization, SC24-5847

VM/ESA: TCP/IP Function Level 320 User's Guide, SC24-5848

VM/ESA: TCP/IP Function Level 320 Programmer's Reference, SC24-5849

VM/ESA: TCP/IP Function Level 320 Messages and Codes, GC24-5850

VM/ESA: TCP/IP Function Level 320 Diagnosis Guide, GC24-5851

OpenEdition Distributed Computing Environment Feature for VM/ESA

IBM OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment, SC24-5735

IBM OpenEdition DCE for VM/ESA: Planning, SC24-5737

IBM OpenEdition DCE for VM/ESA: Configuring and Getting Started, SC24-5734

IBM OpenEdition DCE for VM/ESA: Administration Guide, SC24-5730

IBM OpenEdition DCE for VM/ESA: Administration Reference, SC24-5731

IBM OpenEdition DCE for VM/ESA: Application Development Guide, SC24-5732

IBM OpenEdition DCE for VM/ESA: Application Development Reference, SC24-5733

IBM OpenEdition DCE for VM/ESA: User's Guide, SC24-5738

IBM OpenEdition DCE for VM/ESA: Messages and Codes, SC24-5736

LAN File Services/ESA

Discovering LAN File Services/ESA, GK2T-5762

Introducing LAN File Services/ESA, GH24-5259

LAN File Services/ESA: Licensed Program Specifications, GH24-5260

LAN File Services/ESA: VM Guide and Reference, SH24-5264

LAN Resource Extension and Services/VM

LAN Resource Extension and Services/VM: Licensed Program Specifications, GC24-5617

LAN Resource Extension and Services/VM: General Information, GC24-5618

LAN Resource Extension and Services/VM: Guide and Reference, SC24-5622

CD-ROM

The following CD-ROM contains PDF versions of many VM/ESA publications and publications for some related IBM licensed programs. It also contains all the IBM libraries that are available in IBM BookManager format for current VM system products and current IBM licensed programs that run on VM/ESA.

IBM Online Library Omnibus Edition: VM Collection, SK2T-2067

Note: Only unlicensed publications are included.

Index

Special Characters

A

B

C

D

E

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

Communicating Your Comments to IBM

Virtual Machine/
Enterprise Systems Architecture
Service Guide
Version 2 Release 4.0

Publication No. GC24-5838-01

If there is something you like—or dislike—about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 416-448-6161
  - Other countries: (+1)-416-448-6161
- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you wish a reply.
  - Internet: torrcf@ca.ibm.com
  - IBMLink: toribm(torrcf)
  - IBM/PROFS: torolab4(torrcf)
  - IBMMAIL: ibmmail(caibmwt9)

Readers' Comments — We'd Like to Hear from You

Virtual Machine/
Enterprise Systems Architecture
Service Guide
Version 2 Release 4.0

Publication No. GC24-5838-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
GC24-5838-01

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 EGLINTON AVENUE EAST
NORTH YORK ONTARIO CANADA M3C 1H7

Fold and Tape **Please do not staple** Fold and Tape

GC24-5838-01

Cut or Fold
Along Line

File Number: S370/S390-37
Program Number: 5654-030

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC24-5838-01

Spine information:

VM/ESA Service Guide

Version 2 Release 4.0