

z/VM



Service Guide

version 5 release 3

z/VM



Service Guide

version 5 release 3

Note:

Before using this information and the product it supports, read the information under “Notices” on page L-1.

This edition applies to version 5, release 3, modification 0 of IBM z/VM (product number 5741-A05) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC24-6117-01.

© **Copyright International Business Machines Corporation 1991, 2007. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	xi
Who Should Read This Book	xi
What You Should Know before Reading This Book	xi
What this Book Contains	xi
How To Use This Book	xi
Conventions Used in This book	xi
Where to Find More Information.	xii
How to Send Your Comments to IBM	xii
 Summary of Changes	xv
GC24-6117-02, z/VM Version 5 Release 3	xv
GC24-6117-01, z/VM Version 5 Release 2	xv
GC24-6117-00, z/VM Version 5 Release 1	xv
 Chapter 1. Servicing Your System	1-1
Overview of the Product Service Upgrade (PSU) Procedure	1-1
Overview of the Service Procedure	1-2
Merging Service	1-3
Receiving Service	1-3
Applying Service	1-3
Reworking Local Service (If Applicable)	1-4
Building New Levels.	1-4
Placing the Serviced Components into Production.	1-4
Service Dependencies among Components	1-4
Recommended Order of Service	1-5
Out-of-Component Requisites	1-6
Service Packaging	1-6
Disks Used to Service z/VM.	1-7
VMSES/E Minidisks and Directory IDs	1-8
CMS and REXX/VM Minidisks and Directory IDs	1-9
CP and Dump Viewing Facility Minidisks and Directory IDs	1-10
GCS Minidisks and Directory IDs	1-11
TSAF and AVS Minidisks and Directory IDs	1-12
Language Environment Minidisks and Directory IDs	1-13
Building New Objects.	1-13
Build Tasks by Component.	1-14
Using Tools for Service and System Generation	1-14
Using the Online HELP Facility	1-17
 Chapter 2. Using the Product Service Upgrade (PSU) Procedure	2-1
Materials Needed for a Product Service Upgrade	2-3
Spooling Your Console for non-VMSES messages	2-4
Prepare Your System for Service Refresh.	2-4
Prepare to Receive a Component from the RSU	2-7
Invoke the VMFPSU Command	2-7
Receive a Component from the RSU	2-9
Process Additional Service	2-9
Build the New Service Level	2-10
Service Another Component?	2-11
Process the PSP Bucket COR Tape	2-11
Place Product Service into Production.	2-11
 Chapter 3. Using the COR Service Procedure	3-1

Determine the RSU Service Level Before Ordering COR Service	3-1
How to Install COR Service	3-1
Using Substitute Disks for Default Disks	3-3
Spooling Your Console for non-VMSES messages	3-3
The COR Service Procedure	3-4
Step 1. Prepare the System for Service	3-5
Step 2. Receive the Service Documentation	3-6
Step 3. Service VMSES/E	3-7
Prepare to Receive Service	3-7
Receive the Service.	3-8
Prepare to Apply Service	3-8
Apply the Service.	3-9
Prepare to Build	3-9
Update the Build Status Table	3-10
Build Serviced Objects	3-11
Merge Processing	3-11
Where to Next?	3-12
Step 4. Service REXX/VM	3-13
Prepare to Receive Service	3-13
Receive the Service	3-14
Prepare to Apply Service	3-14
Apply the Service	3-15
Prepare to Build.	3-16
Update the Build Status Table	3-16
Build Serviced Objects	3-17
Merge Processing	3-18
Where to Next?	3-18
Step 5. Service Language Environment	3-19
Prepare to Receive Service	3-19
Receive the Service	3-20
Prepare to Apply Service	3-20
Apply the Service	3-21
Prepare to Build.	3-21
Update the Build Status Table	3-22
Build Serviced Objects	3-23
Merge Processing	3-23
Where to Next?	3-24
Step 6. Service CMS	3-25
Prepare to Receive Service	3-25
Receive the Service	3-26
Prepare to Apply Service	3-26
Apply the Service	3-27
Prepare to Build.	3-28
Update the Build Status Table	3-28
Build Serviced Objects	3-29
Merge Processing	3-32
Where to Next?	3-33
Step 7. Service CP	3-34
Prepare to Receive Service	3-34
Receive the Service	3-35
Prepare to Apply Service	3-35
Apply the Service	3-36
Prepare to Build.	3-37
Update the Build Status Table	3-37
Build Serviced Objects	3-38
Merge Processing	3-40

Where to Next?	3-41
Step 8. Service GCS	3-42
Prepare to Receive Service	3-42
Receive the Service	3-43
Prepare to Apply Service	3-43
Apply the Service	3-44
Prepare to Build.	3-45
Update the Build Status Table	3-45
Build Serviced Objects	3-46
IPL the GCS Nucleus.	3-47
Merge Processing	3-48
Where to Next?	3-48
Step 9. Service the Dump Viewing Facility	3-49
Prepare to Receive Service	3-49
Receive the Service	3-50
Prepare to Apply Service	3-50
Apply the Service	3-51
Prepare to Build.	3-52
Update the Build Status Table	3-52
Build Serviced Objects	3-53
Merge Processing	3-53
Where to Next?	3-54
Step 10. Service TSAF	3-55
Prepare to Receive Service	3-55
Receive the Service	3-56
Prepare to Apply Service	3-56
Apply the Service	3-57
Prepare to Build.	3-57
Update the Build Status Table	3-58
Build Serviced Objects	3-59
Merge Processing	3-59
Where to Next?	3-60
Step 11. Service AVS.	3-61
Prepare to Receive Service	3-61
Receive the Service	3-62
Prepare to Apply Service	3-62
Apply the Service	3-63
Prepare to Build.	3-64
Update the Build Status Table	3-64
Build Serviced Objects	3-65
Merge Processing	3-65
Where to Next?	3-66
Step 12. Place Your Serviced Components into Production	3-67
Rebuild the Saved Segments	3-67
Move Files into the Byte File System	3-71
Place the Test Build Disks into Production	3-72
Rebuild the CMS Saved System.	3-73
Place the CP Nucleus into Production.	3-73
Place TSAF and AVS into Production	3-75
Place GCS into Production.	3-75
Merge Tested Service Level	3-75
Chapter 4. Detailed Component Build Steps	4-1
Detailed Build Steps: VMSES/E	4-1
Prepare to Build	4-1
Determine Which Build Lists Need to be Processed	4-1

Select and Copy Serviced Files	4-1
Rebuild the Executable MODULES	4-2
Process the Segment Build List	4-2
Access the New Level of VMSES/E	4-3
Detailed Build Steps: REXX/VM	4-4
Prepare to Build	4-4
Determine Which Build Lists Need to be Processed	4-4
Select and Copy Serviced Files	4-4
Update the Macro Libraries (MACLIBs).	4-5
Rebuild the Nuclei	4-5
Detailed Build Steps: Language Environment	4-6
Prepare to Build	4-6
Determine Which Build Lists Need to be Processed	4-6
Select and Copy Serviced Files	4-7
Update the Macro Libraries (MACLIBs).	4-8
Rebuild the Text Libraries.	4-8
Rebuild the Load Libraries	4-9
Rebuild the Executable MODULES	4-9
Process the Segment Build Lists	4-10
Detailed Build Steps: CMS	4-12
Prepare to Build.	4-12
Determine Which Build Lists Need to be Processed	4-12
Select and Copy Serviced Files	4-13
Update the Macro Libraries (MACLIBs)	4-14
Rebuild the Text Libraries	4-15
Rebuild the Load Library	4-15
Rebuild the Executable MODULES	4-16
Rebuild the Assemble Module	4-17
Rebuild the Callable Services Libraries (CSLs)	4-17
Process the Segment Build Lists	4-18
Rebuild the CMS Nucleus	4-19
IPL the CMS Nucleus.	4-19
Detailed Build Steps: CP	4-22
Prepare to Build.	4-22
Determine Which Build Lists Need to be Processed	4-22
Select and Copy Serviced Files	4-23
Update the Macro Libraries	4-23
Rebuild the Executable MODULES	4-24
Process the Segment Build Lists	4-24
Rebuild the CP Utilities	4-25
Rebuild the CP Nucleus.	4-25
Verify the CP Load Map.	4-26
Detailed Build Steps: GCS	4-28
Prepare to Build.	4-28
Determine Which Build Lists Need to be Processed	4-28
Select and Copy Serviced Files	4-28
Update the Macro Libraries	4-29
Rebuild the Load Library	4-29
Process the Segment Build List	4-30
Rebuild the GCS Nucleus	4-30
IPL the GCS Nucleus.	4-31
Detailed Build Steps: Dump Viewing Facility	4-33
Prepare to Build.	4-33
Determine Which Build Lists Need to be Processed	4-33
Select and Copy Serviced Files	4-33
Rebuild the Executable MODULES	4-34

Process the Segment Build List	4-34
Detailed Build Steps: TSAF	4-36
Prepare to Build	4-36
Determine Which Build Lists Need to be Processed	4-36
Select and Copy Serviced Files	4-36
Rebuild the Executable MODULEs	4-37
Process the Segment Build List	4-37
Detailed Build Steps: AVS	4-38
Prepare to Build	4-38
Determine Which Build Lists Need to be Processed	4-38
Select and Copy Serviced Files	4-38
Rebuild the Load Libraries	4-39
Process the Segment Build List	4-39

Chapter 5. Local Service and Modifications — Overview	5-1
Introduction	5-1
Overview for Local Service Procedure	5-2
Overview for Rework Local Service Procedure	5-3
Definitions for Variables	5-3
Obtaining File Type Abbreviations.	5-5
Default Control, Apply, and AUX File Identifiers	5-5

Chapter 6. Procedures for Local Service and Modifications	6-1
Step 1. Prepare for Local Service or Modifications	6-1
Step 2. Receive Local Service or Modifications	6-2
Step 3. Apply Local Service to Source-Maintained Parts	6-3
Step 3a. Add an Update Record to the AUX File	6-4
Step 3b. Create or Copy the Update File	6-4
Step 3c. Special Processing for Macros with ASSEMBLE Files	6-5
Step 4. Apply Local Service to Replacement-Maintained Parts	6-6
Step 4a. Create or Copy the Replacement Part.	6-6
Step 4b. Special Processing for Macros with ASSEMBLE files	6-7
Step 5. Rebuild Objects	6-7
Step 5a. Rebuild Source Files	6-8
Step 5b. Create Replacement Part from Modified \$Source, Compiled REXX	6-8
Step 5c. Create Replacement Part from Modified \$Source, Not Compiled REXX	6-9
Step 5d. Rebuild MACLIBs.	6-10
Step 5e. Create a Replacement Part from Updated ASSEMBLE Files	6-10
Step 5f. Create a Replacement Part from Replaced ASSEMBLE Files	6-11
Step 5g. Create a Replacement Part from Modified National Language Files	6-12
Step 5h. Rebuild Remaining Objects	6-12

Chapter 7. Reworking Local Service and Modifications	7-1
Step 1. Prepare to Rework Local Service or Modifications.	7-1
Step 2. Rework Local Service to Source-Maintained Parts.	7-2
Step 3. Rework Local Service to Replacement-Maintained Parts	7-3
Step 4. Rebuild Objects	7-4
Step 4a. Rebuild Source Files	7-4
Step 4b. Create Replacement Part from Modified \$Source, Compiled REXX	7-4
Step 4c. Create Replacement Part from Modified \$Source, Not Compiled REXX	7-5
Step 4d. Rebuild MACLIBs	7-6
Step 4e. Create a Replacement Part from Updated ASSEMBLE Files	7-6
Step 4f. Create a Replacement Part from Replaced ASSEMBLE Files	7-7

Step 4g. Create a Replacement Part from Modified National Language Files	7-8
Step 4h. Rebuild Remaining Objects	7-8
Chapter 8. Removing Service	8-1
Removing Service Levels.	8-1
Removing A Single PTF	8-2
Restoring Removed Service.	8-2
Removing A Local Modification.	8-3
Appendix A. Receiving Service for VMSES Envelopes	A-1
Determine the Receive Steps to Follow	A-1
Receive Service Documentation	A-1
Receive the Service.	A-2
Appendix B. Receiving Service For Non-VMSES Envelopes	B-1
Appendix C. Building the NLS Nuclei, Including UCENG	C-1
Appendix D. Test the New Level of CMS	D-1
Appendix E. Local Modification Example for CMSINST	E-1
Appendix F. Updating the CP Load List.	F-1
Steps to Follow	F-2
Appendix G. Using VMFBLD Outside of the Service Process	G-1
Appendix H. Apply Changes Directly to Object Code	H-1
Appendix I. Applying and Removing Patches	I-1
Creating a Patch	I-1
Removing a Patch	I-2
Appendix J. z/VM Build Lists.	J-1
Appendix K. Determine the Component Service Level.	K-1
Determine the CMS Service Level	K-1
Determine the CP Service Level	K-1
Determine the GCS Service Level	K-1
Notices	L-1
Trademarks	L-3
Glossary	M-1
Bibliography	N-1
Where to Get z/VM Publications	N-1
z/VM Base Library	N-1
Overview	N-1
Installation, Migration, and Service	N-1
Planning and Administration.	N-1
Customization and Tuning	N-1
Operation and Use	N-1
Application Programming.	N-2
Diagnosis	N-2
Publications for z/VM Optional Features	N-2

Data Facility Storage Management Subsystem for VM	N-2
Directory Maintenance Facility for z/VM	N-2
Performance Toolkit for VM	N-3
RACF Security Server for z/VM	N-3
RSCS Networking for z/VM	N-3
Index	X-1

About This Book

This book describes how to service version 5 release 3 of IBM® z/VM®.

Who Should Read This Book

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

What You Should Know before Reading This Book

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

What this Book Contains

This book contains the Product Service Upgrade (PSU) service procedure, which utilizes the z/VM Recommended Service Upgrade (RSU), and the corrective (COR) service procedure for all z/VM components. Local service and modifications as well as reworking local service is also discussed. The appendices include supplemental information and procedures for service. Legal notices, a reference to the z/VM glossary, a bibliography, and an index are also included in this book.

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.

Conventions Used in This book

Substeps in procedures display what you should type and the system responses you see.

- Bold font indicates exactly what you should type.

Example: The following shows a command you would type:

disconnect

- Normal font indicates system responses and requests.

Example: The following shows a system response:

```
HCPIPX8475I THE PRODUCTS YOU SELECTED TO LOAD TO MINIDISK ARE:
              VM RSCS TCPIP OSA ICKDSF
              DIRM RACF PERFTK VMHCD
```

- Italic font indicates variable input or output, which can occur in commands you type or in system output.

Examples: The following are examples in which italics indicate variable input or output:

- In the following, you would need to supply the address of a tape drive for *tapeaddr*:

attach *tapeaddr* * 181

- In the following, the system would supply a tape address for *tapeaddr* and *userID* in its response:

TAPE *tapeaddr* ATTACHED TO *userID* 181

- Reverse type indicates special keys you must press.

Example: The following indicates you must press the Enter key:

ENTER

- Explanatory notes appear to the right of input and output.

Example:

attach *tapeaddr* * 181

TAPE *tapeaddr* ATTACHED TO *userID* 181
Ready; T=*n.nn/n.nn hh:mm:ss*

tapeaddr is the address of the tape drive(s) where the z/VM System DDR tapes will be mounted. *userID* is the first-level user ID logged on to in the previous substep.

Where to Find More Information

See the “Bibliography” on page N-1 for more information concerning z/VM.

For more information on the service tools provided by VMSES/E (besides reviewing the *z/VM: VMSES/E Introduction and Reference*) use the online z/VM 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 *z/VM: CMS User's Guide*. To display the main HELP Task Menu, enter:

help

For more information about the HELP command, see the *z/VM: CMS Commands and Utilities Reference* or enter:

help cms help

Links to Other Online Books

If you are viewing the Adobe® Portable Document Format (PDF) version of this book, it may contain links to other books. A link to another book is based on the name of the requested PDF file. The name of the PDF file for an IBM book is unique and identifies the edition. The links provided in this book are for the editions (PDF names) that were current when the PDF file for this book was generated. However, newer editions of some books (with different PDF names) may exist. A link from this book to another book works only when a PDF file with the requested name resides in the same directory as this book.

How to Send Your Comments to IBM

IBM welcomes your comments. You can use any of the following methods:

- Complete and mail the Readers' Comments form (if one is provided at the back of this book) or send your comments to the following address:

IBM Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, New York 12601-5400
U.S.A.

- Send your comments by FAX:
 - United States and Canada: 1-845-432-9405
 - Other Countries: +1 845 432 9405
- Send your comments by electronic mail to one of the following addresses:
 - Internet: mhvrcfs@us.ibm.com
 - IBMLink™ (US customers only): IBMUSM10(MHVRCFS)

Be sure to include the following in your comment or note:

- Title and complete publication number of the book
- Page number, section title, or topic you are commenting on

If you would like a reply, be sure to also include your name, postal or e-mail address, telephone number, or FAX number.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to left of the change.

GC24-6117-02, z/VM Version 5 Release 3

This edition supports the general availability of z/VM V5.3.

GC24-6117-01, z/VM Version 5 Release 2

This edition supports the general availability of z/VM V5.2.

GC24-6117-00, z/VM Version 5 Release 1

This edition supports the general availability of z/VM V5.1.

- Only one CP nucleus is built, which is a 64 bit nucleus.

Chapter 1. Servicing Your System

The preferred method of installing service is to use the new automated service procedure documented in *z/VM: Guide for Automated Installation and Service*. If you choose, you can still use the traditional service procedure documented in this document.

This book contains procedures you can use to install the following kinds of service to z/VM:

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

Notes:

1. The preventive and corrective service, the local service, and local modification procedures in this book have been automated by the SERVICE, PUT2PROD and LOCALMOD commands. The procedures for using these commands are documented in *z/VM: Guide for Automated Installation and Service*.
2. This book is designed to be used with the *z/VM: VMSES/E Introduction and Reference*.
3. *z/VM: VMSES/E Introduction and Reference* discusses the concepts of z/VM 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 *z/VM: VMSES/E Introduction and Reference* ("Chapter 1. Introducing VMSES/E" and "Chapter 8. z/VM Service Concepts" would be recommended) before you begin using the procedures in this book.
4. Also, from time to time, this book points to reference information contained in *z/VM: CP Planning and Administration*, *z/VM: CMS Planning and Administration*, and *z/VM: Saved Segments Planning and Administration*.

Overview of the Product Service Upgrade (PSU) Procedure

The PSU procedure is the method of installing preventive maintenance to z/VM version 5 release 3.

The Product Service Upgrade (PSU) is a procedure that refreshes the service database disks defined for z/VM version 5 release 3, leaves customized files untouched, and preserves any reach-ahead service. This procedure utilizes the z/VM 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 z/VM version 5 release 3 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, CMS nucleus, and GCS nucleus must be rebuilt because they are not included on the tapes.

Overview of the Service Procedure

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

1. Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E)
2. REXX/VM
3. Language Environment®
4. Conversational Monitor System (CMS)
5. Control Program (CP)
6. Group Control System (GCS)
7. Dump Viewing Facility
8. Transparent Services Access Facility (TSAF)
9. APPC/VM VTAM® Support (AVS)

When you service a z/VM 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.

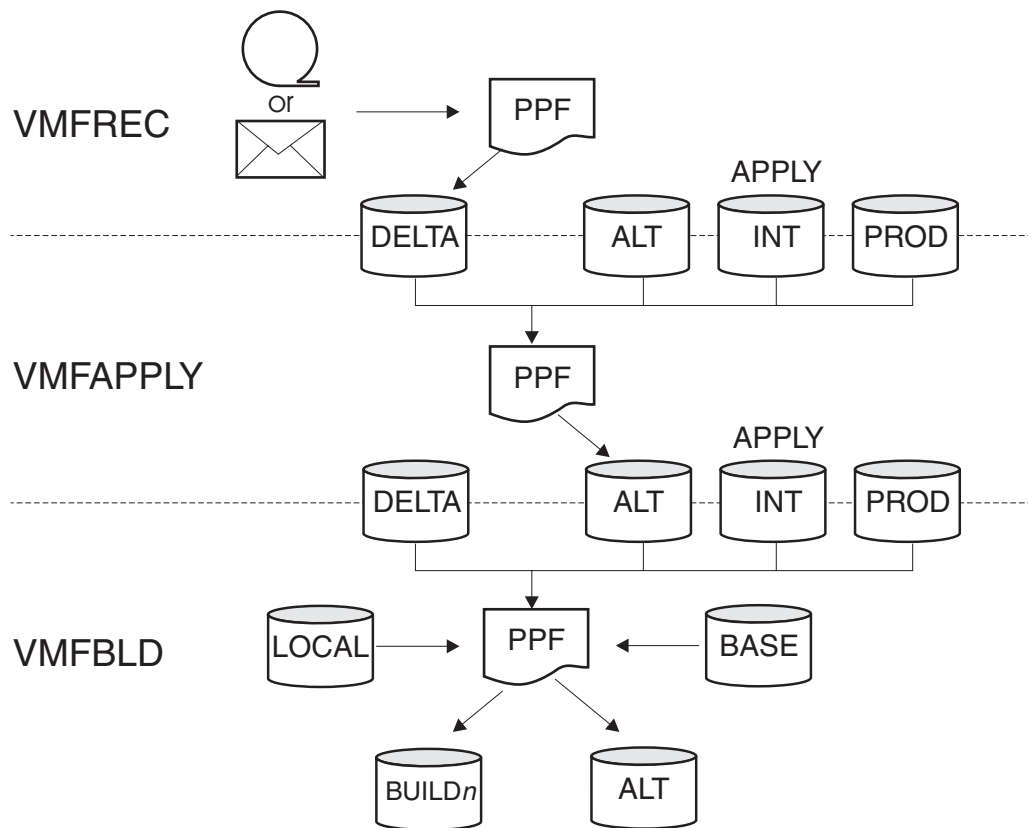


Figure 1-1. Disks used when installing service

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 *z/VM: 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

Servicing Your System

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.

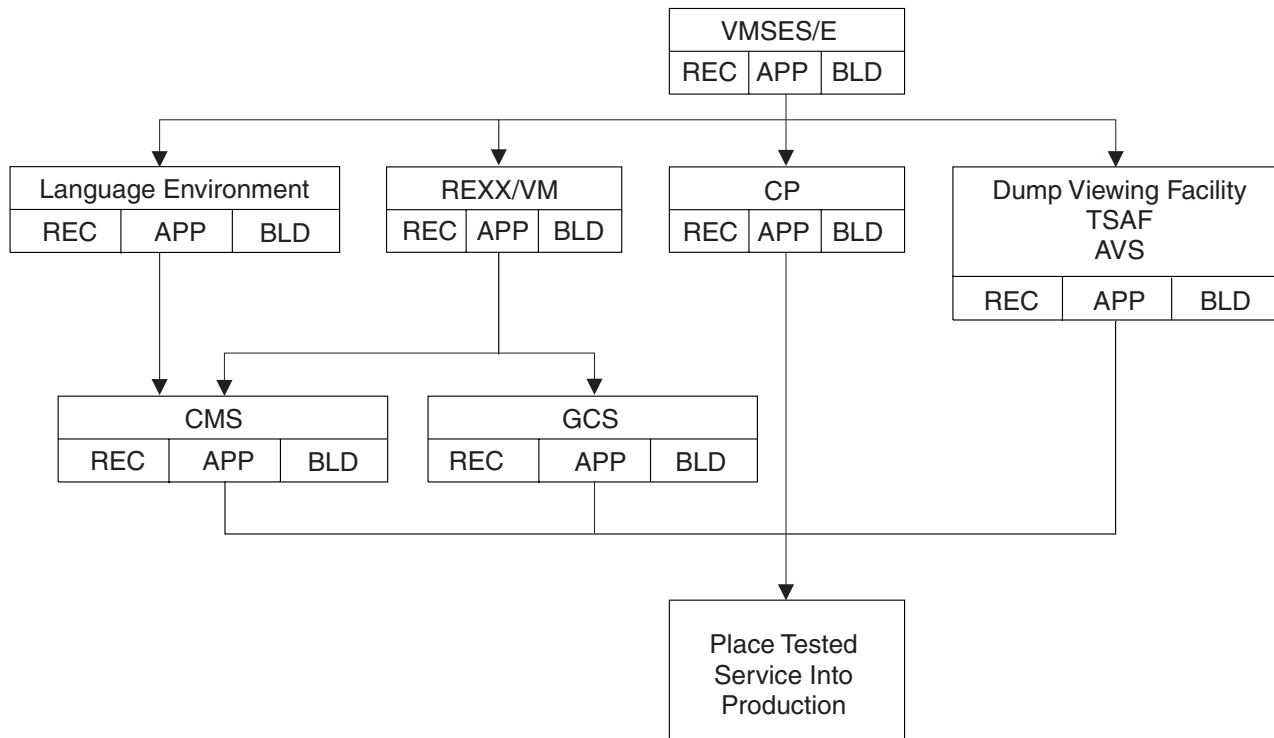


Figure 1-2. Component dependencies for installing service

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. 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. Because Language Environment is used in building some CMS modules, you must receive, apply, and build Language Environment before you build CMS.

Recommended Order of Service

Because of these dependencies, IBM recommends that you service z/VM 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 z/VM 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. Language Environment—Receive, Apply, and Build

4. CMS—Receive, Apply, and Build

After REXX/VM service has been received and applied, and Language Environment service has been received, applied, and built, CMS can be processed. REXX/VM service must be received and applied before you build the CMS nucleus.

Servicing Your System

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.

5. CP—Receive, Apply, and Build
6. 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.

7. 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

z/VM 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 (ZVM \$PPF) is provided for z/VM 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 lists the file names of the base \$PPFs, descriptions of each file, and their component names as defined in the ZVM \$PPF file.

Table 1-1. z/VM Component PPF file names and aliases

Base \$PPF File Name Description		Component Name as Defined in ZVM \$PPF	
		Installed on minidisks	Installed in the shared file system
5VMSES30	VMSES/E Product Parameter File	VMSES	VMSESSFS
5VMREX30	REXX/VM Product Parameter File	REXX	REXXSFS
4VMVMQ40	Language Environment Product Parameter File	LE	LESFS
5VMCMS30	CMS Product Parameter File	CMS	CMSSFS
5VMCPR30	CP Product Parameter File	CP	CPSFS
5VMGCS30	GCS Product Parameter File	GCS	GCSSFS

Table 1-1. z/VM Component PPF file names and aliases (continued)

Base \$PPF File Name	Description	Component Name as Defined in ZVM \$PPF	
		Installed on minidisks	Installed in the shared file system
5VMDVF30	Dump Viewing Facility Product Parameter File	DV	DVSFS
5VMTSA30	TSAF Product Parameter File	TSAF	TSAFSFS
5VMAVS30	AVS Product Parameter File	AVS	AVSSFS

Disks Used to Service z/VM

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. VMSES/E, Language Environment, 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 z/VM version 5 release 3.

LOCALMOD

The local disk (*nC4*) contains any local modifications you have installed, except PTF fixes.

LOCALSAM

The local sample disk (*nC2*) contains the sample and example files you receive from IBM; for example, LOGO CONFIG and DMSNGP.

DELTA

The delta disk (*nD2*) 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 z/VM.

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 z/VM: the alternate apply disk, the intermediate apply disk, and the production apply disk.

Alternate Apply disk (*nA6*)

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 (*nA4*)

This disk contains the level of service that was last installed. The

Servicing Your System

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 (*nA2*)

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 z/VM. Note that some of the components share the same disks for service, such as CP and the Dump Viewing Facility.

VMSES/E Minidisks and Directory IDs

VMSES/E Minidisks

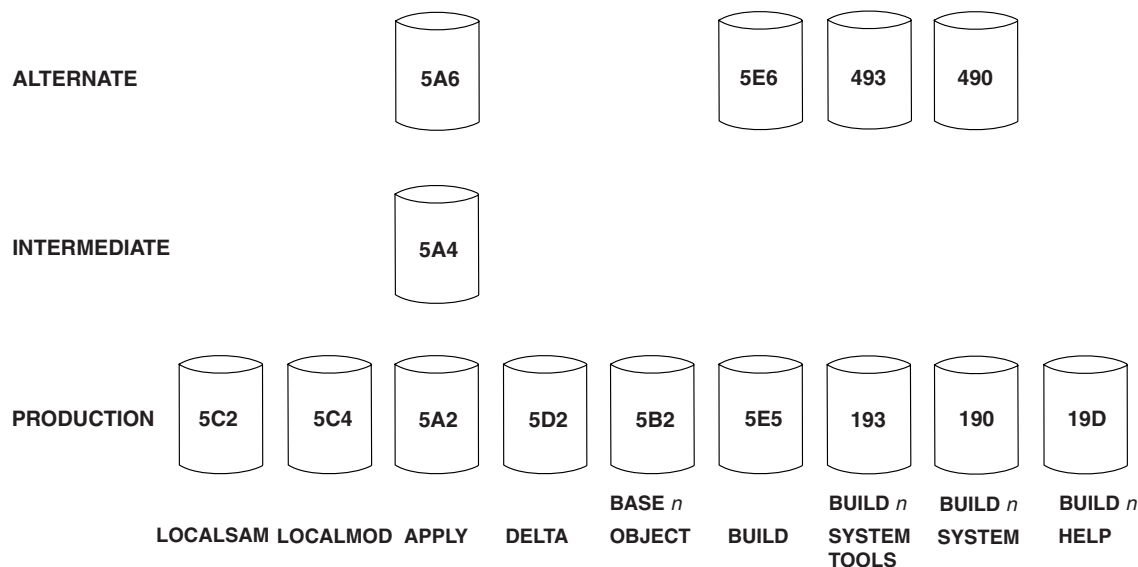


Figure 1-3. Disks used when servicing VMSES/E

VMSES/E Directory IDs

Note: If you install z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-2. VMSES/E minidisks and comparable directory IDs

Minidisk	Directory ID
5A2	VMSYS:MAINT.VMSES.APPLYPROD
5A4	VMSYS:MAINT.VMSES.APPLYINT
5A6	VMSYS:MAINT.VMSES.APPLYALT
5D2	VMSYS:MAINT.VMSES.DELTAPROD
5B2	VMSYS:MAINT.VMSES.OBJECT
5C2	VMSYS:MAINT.VMSES.SAMPLE
5C4	VMSYS:MAINT.VMSES.LOCALMOD

CMS and REXX/VM Minidisks and Directory IDs

CMS and REXX/VM Minidisks

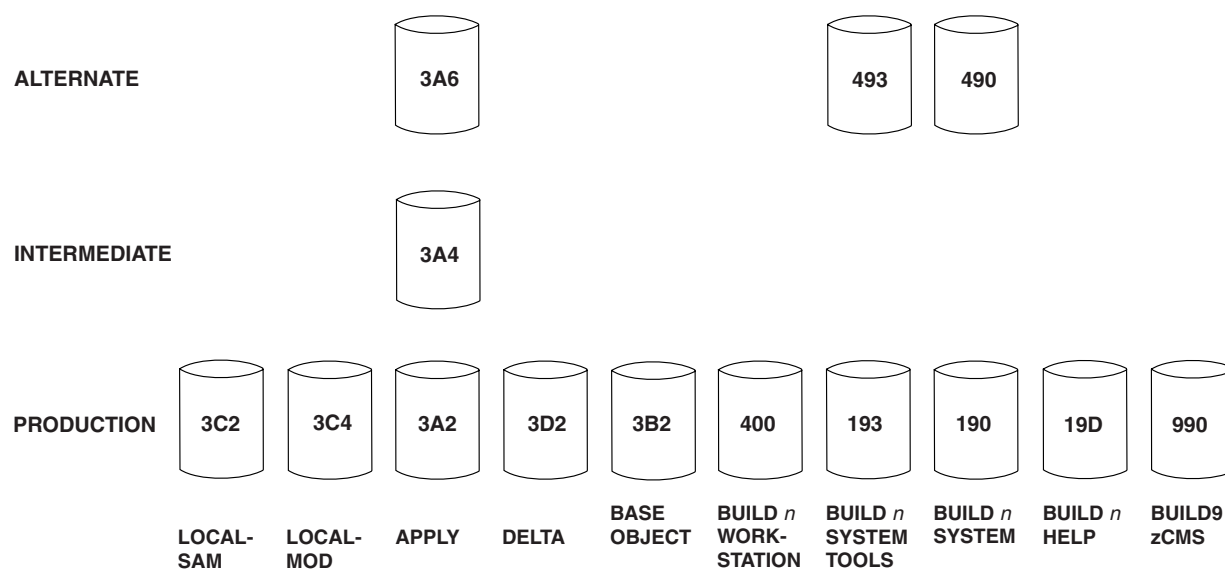


Figure 1-4. Disks used when servicing CMS and REXX/VM

CMS and REXX/VM Directory IDs

Note: If you install z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-3. CMS and REXX/VM minidisks and comparable directory IDs

Minidisk	Directory ID
3A2	VMSYS:MAINT.CMSREXX.APPLYPROD
3A4	VMSYS:MAINT.CMSREXX.APPLYINT
3A6	VMSYS:MAINT.CMSREXX.APPLYALT
3D2	VMSYS:MAINT.CMSREXX.DELTAPROD
3B2	VMSYS:MAINT.CMSREXX.OBJECT
3C2	VMSYS:MAINT.CMSREXX.SAMPLE
3C4	VMSYS:MAINT.CMSREXX.LOCALMOD

CP and Dump Viewing Facility Minidisks and Directory IDs

CP and Dump Viewing Facility Minidisks

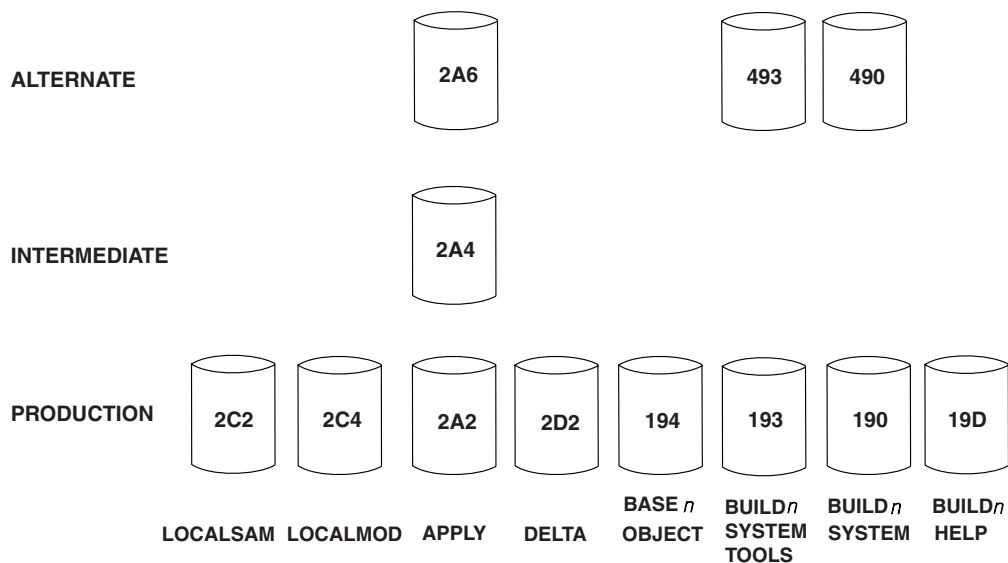


Figure 1-5. Disks used when servicing CP and Dump Viewing Facility

CP and Dump Viewing Facility Directory IDs

Note: If you install z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-4. CP and Dump Viewing Facility minidisks and comparable directory IDs

Minidisk	Directory ID
2A2	VMSYS:MAINT.CPDV.APPLYPROD
2A4	VMSYS:MAINT.CPDV.APPLYINT
2A6	VMSYS:MAINT.CPDV.APPLYALT
2D2	VMSYS:MAINT.CPDV.DELTAPROD
194	VMSYS:MAINT.CPDV.OBJECT
2C2	VMSYS:MAINT.CPDV.SAMPLE
2C4	VMSYS:MAINT.CPDV.LOCALMOD

GCS Minidisks and Directory IDs

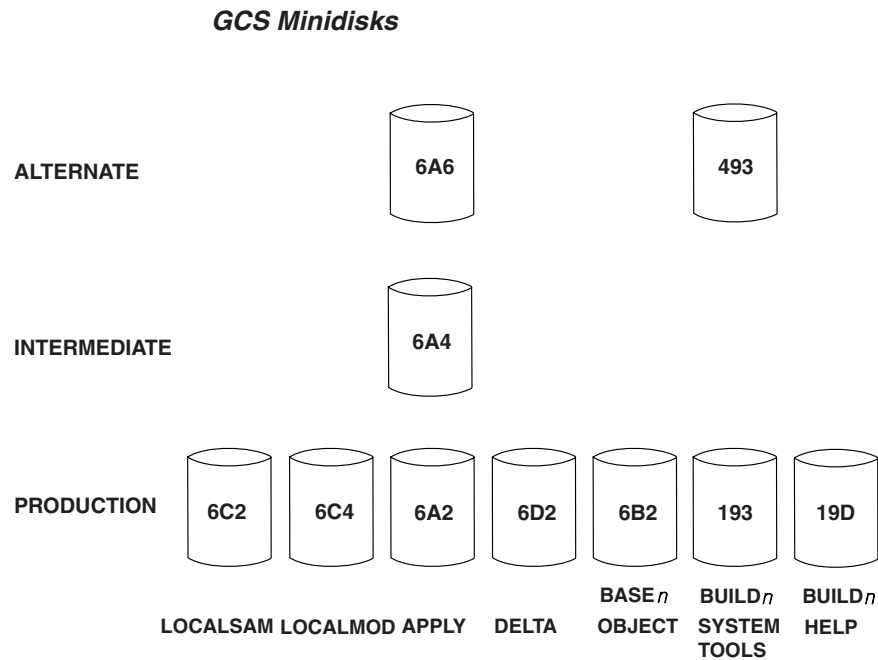


Figure 1-6. Disks used when servicing GCS

GCS Directory IDs

Note: If you are installing z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-5. 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
6C2	VMSYS:MAINT.GCS.SAMPLE
6C4	VMSYS:MAINT.GCS.LOCALMOD

TSAF and AVS Minidisks and Directory IDs

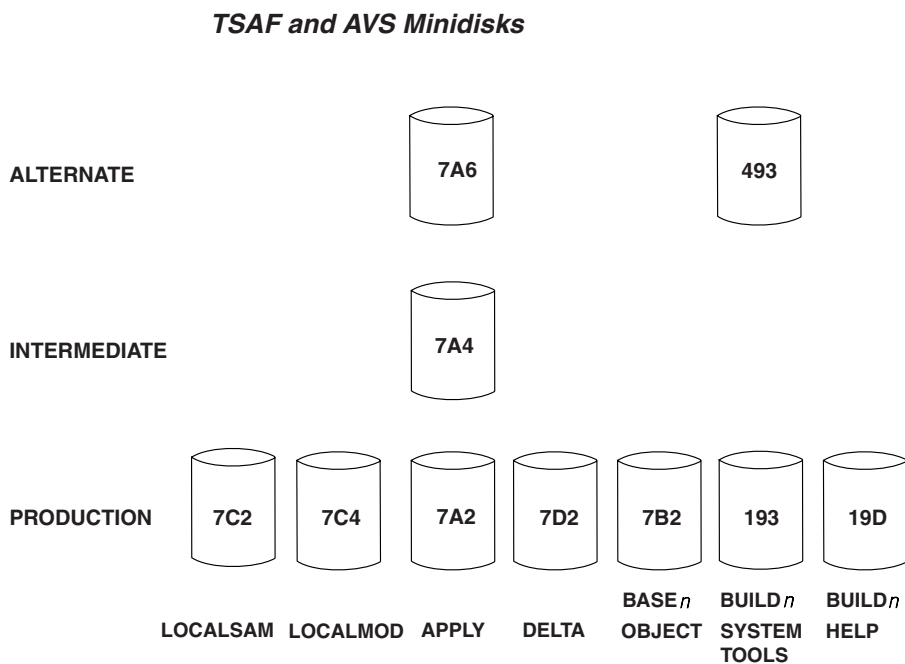


Figure 1-7. Disks used when servicing TSAF and AVS

TSAF and AVS Directory IDs

Note: If you are installing z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-6. TSAF and AVS minidisks and comparable directory IDs

Minidisk	Directory ID
7A2	VMSYS:MAINT.TSAFAVS.APPLYPROD
7A4	VMSYS:MAINT.TSAFAVS.APPLYINT
7A6	VMSYS:MAINT.TSAFAVS.APPLYALT
7D2	VMSYS:MAINT.TSAFAVS.DELTAPROD
7B2	VMSYS:MAINT.TSAFAVS.OBJECT
7C2	VMSYS:MAINT.TSAFAVS.SAMPLE
7C4	VMSYS:MAINT.TSAFAVS.LOCALMOD

Language Environment Minidisks and Directory IDs

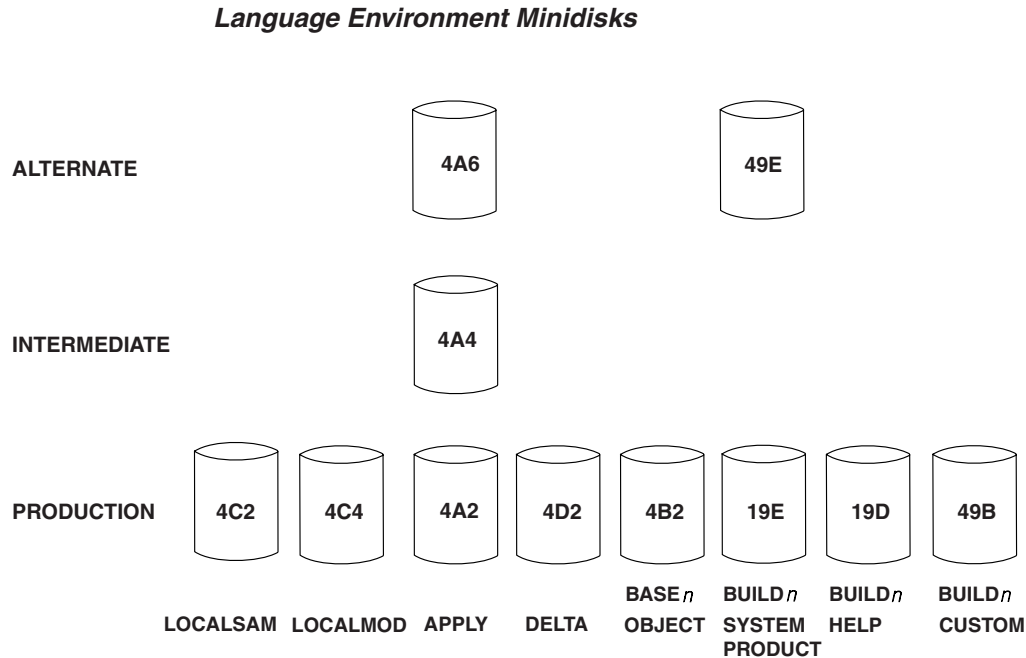


Figure 1-8. Disks used when servicing Language Environment

Language Environment Directory IDs

Note: If you are installing z/VM to the shared file system (SFS), then the following directory IDs match the minidisk definitions.

Table 1-7. Language Environment minidisks and comparable directory IDs

Minidisk	Directory ID
4A2	VMSYS:MAINT.LE.APPLYPROD
4A4	VMSYS:MAINT.LE.APPLYINT
4A6	VMSYS:MAINT.LE.APPLYALT
4D2	VMSYS:MAINT.LE.DELTAPROD
4B2	VMSYS:MAINT.LE.OBJECT
4C2	VMSYS:MAINT.LE.SAMPLE
4C4	VMSYS:MAINT.LE.LOCALMOD

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

Servicing Your System

z/VM 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 1-8 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 1-8. Build tasks for z/VM components

VMSES/E	REXX/VM	Language Environment	CMS	CP	GCS	Dump Viewing Facility	TSAF	AVS
Update \$Source Files								
VMFEXUPD	VMFEXUPD	NA	VMFEXUPD	VMFEXUPD	NA	VMFEXUPD	NA	NA
Rebuild MACLIBs								
NA	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	NA	NA	NA
Reassemble Source Files								
NA	VMFHLASM	VMFHLASM	VMFHLASM	VMFHLASM	NA	NA	NA	NA
Rebuild NLS Parts								
VMFNLS	NA	VMFNLS	VMFNLS	VMFNLS	VMFNLS	VMFNLS	VMFNLS	VMFNLS
Select and Copy Serviced Files								
VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD	VMFBLD
Rebuild TXTLIBs								
NA	NA	VMFBLD	VMFBLD	NA	NA	NA	NA	NA
Rebuild Modules								
VMFBLD	NA	VMFBLD	VMFBLD	VMFBLD	NA	VMFBLD	VMFBLD	NA
Rebuild LOADLIBs								
NA	NA	VMFBLD	VMFBLD	NA	VMFBLD	NA	NA	VMFBLD
Rebuild CSLLIBs								
NA	NA	NA	VMFBLD	NA	NA	NA	NA	NA
Rebuild the F Assembler								
NA	NA	NA	VMFBLD	NA	NA	NA	NA	NA
Rebuild Nuclei								
NA	NA	NA	VMFBLD	VMFBLD	VMFBLD	NA	NA	NA
Rebuild Saved Segments								
NA	NA	VMFBLD	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

z/VM provides a number of tools to help you perform service and system generation tasks. Table 1-9 on page 1-15 lists z/VM service and system generation commands and the books describing each. Use the following key for this table.

Abbreviation

VMSES/E I and R

Title

z/VM: VMSES/E Introduction and Reference

CMS Cmd Ref

z/VM: CMS Commands and Utilities Reference

CP Cmd Ref

z/VM: CP Commands and Utilities Reference

GCS Ref

z/VM: Group Control System

Table 1-9. z/VM Installation, 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
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
INSTFPP	Installs optional products.	VMSES/E I and R
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
LOCALMOD	Creates local modifications.	VMSES/E I and R
PRELOAD	Collects multiple text files and reformats them into a single text file.	CMS Cmd Ref
PUT2PROD	Places the z/VM components, features, or products that are installed on the z/VM System DDR into production.	VMSES/E I and R
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 discontinuous shared segment (DCSS).	CMS Cmd Ref
SEGEN	Builds logical saved segments defined in a physical saved segment.	CMS Cmd Ref
SERVICE	Installs an RSU or applies CORrective service.	VMSES/E I and R
SNTINFO	Gets discontinuous 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

Servicing Your System

Table 1-9. z/VM Installation, service, and system generation tools (continued)

Tool	Task	Book
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
VMFHASHM	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 COR service (or both) from envelope files or tape.	VMSES/E I and R
VMFSUFTB	Builds the VM SYSSUF table for use by Service Update Facility/390.	VMSES/E I and R

Table 1-9. z/VM Installation, service, and system generation tools (continued)

Tool	Task	Book
VMFTXT	Builds a text library (TXTLIB) from text decks.	CMS Cmd Ref
VMFUPDAT	Updates selected software inventory tables.	VMSES/E I and R
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
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 z/VM 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 *z/VM: CMS User's Guide*. To display the main HELP Task Menu, enter:

```
help
```

For more information about the HELP command, see the *z/VM: CMS Commands and Utilities Reference* or enter:

```
help cms help
```


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

You can upgrade your existing z/VM version 5 release 3 system using the service files from the IBM z/VM Recommended Service Upgrade. **This is the method of installing preventive service to VM** that was introduced in VM/ESA® Release 1.1. The procedure for performing a Product Service Upgrade from the z/VM 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 z/VM version 5 release 3.

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

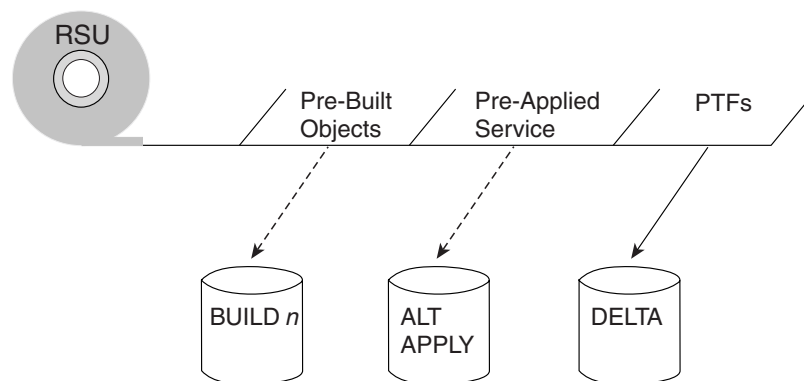


Figure 2-1. RSU and Files

The PSU procedure installs 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.

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.
- The CMS and GCS nuclei, as well as saved segments, need to be rebuilt to include the new service.

Using the PSU Procedure

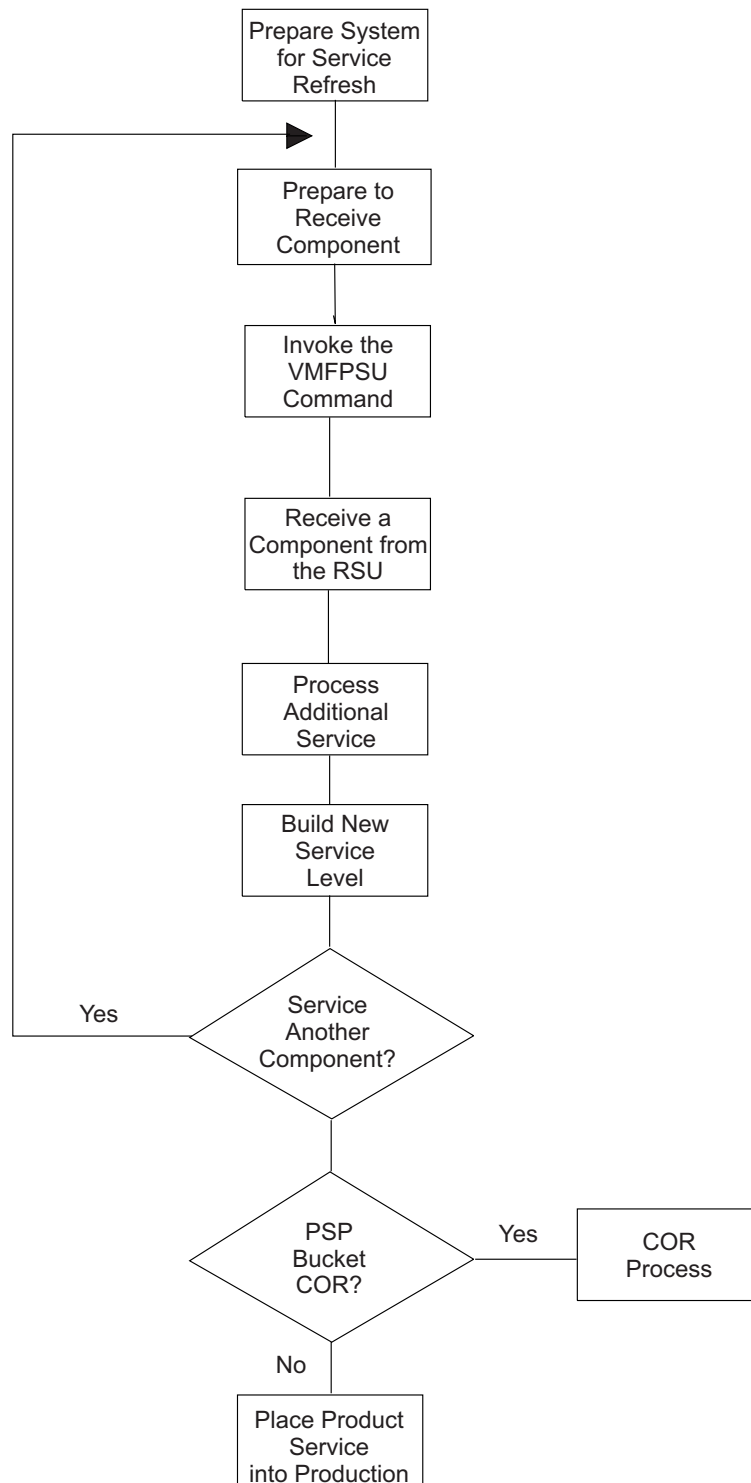


Figure 2-2. Service Application Flowchart using PSU

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 2-4 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 - 7) should be performed for each component in z/VM, 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 2-7, you will set up your environment.

3. 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.

4. Receive a Component from the RSU

Receive Preapplied, Prebuilt Service for a component. Page 2-9 contains the receive process for the RSU.

5. Process Additional Service for a Component

"Process Additional Service" on page 2-9 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.

6. 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 2-10 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.

7. Service Another Component?

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

8. 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.

9. 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 z/VM RSU:
You need to order the z/VM RSU. You can order the RSU through the IBM Support Center.
- In addition to this book, you will also need the following documentation:
 - *z/VM: Guide for Automated Installation and Service*, supplied with your initial product order
 - *z/VM: VMSES/E Introduction and Reference*
- Also, contact your support center for the latest installation information, for example, the RSU PSP bucket.

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 STEP CLOSE`.

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 *z/VM: VMSES/E Introduction and Reference*.

2. If necessary, access the required disks for the service procedure.

`access vdev a`
`access 5e5 b`
`access 51d d`

`vdev` is the virtual address of your R/W A-disk.
Access the VMSES/E installation and service tools disk.
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 that you received electronically, in a README file or in
| the *ServiceLink Users' Guide*, to retrieve and decompress the envelope file on
| your A-disk. You will run DETERSE to decompress the envelope file. The file
| name format for the decompressed file will be `RPTFnum`. The file type will be
| SERVLINK. You will need to enter the file name 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 VVT$PSU$`), 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 rptfnum`

Use `ENV rptfnum` 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 z/VM 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 and can be written in the column Required 4K Blocks in the table below for reference.

Table 2-1. Component Name and Product Identifier

<i>compname</i>	<i>prodid</i> and default <i>appid</i> Identifier	DELTA/APPLY Disks or Directories	Required 4K Blocks (see <i>prodid</i> MEMO)
VMSES	5VMSES30	5D2/5A6	
VMSESSFS	5VMSES30	.vmstes.deltaprod/ .vmstes.applyalt	
REXX	5VMREX30	3D2/3A6	
REXXSFS	5VMREX30	.cmsrex.deltaprod/ .cmsrex.applyalt	
LE	4VMVMQ40	4D2/4A6	
LESFS	4VMVMQ40	.le.deltaprod/ .le.applyalt	
CMS	5VMCMS30	3D2/3A6	
CMSSFS	5VMCMS30	.cmsrex.deltaprod/ .cmsrex.applyalt	
CP	5VMCPR30	2D2/2A6	
CPSFS	5VMCPR30	.cpdv.deltaprod/ .cpdv.applyalt	
GCS	5VMGCS30	6D2/6A6	
GCSSFS	5VMGCS30	.gcs.deltaprod/ .gcs.applyalt	
DV	5VMDVF30	2D2/2A6	
DVSFS	5VMDVF30	.cpdv.deltaprod/ .cpdv.applyalt	
TSAF	5VMTSA30	7D2/7A6	
TSAFSFS	5VMTSA30	.tsafavs.deltaprod/ .tsafavs.applyalt	
AVS	5VMAVS30	7D2/7A6	
AVSSFS	5VMAVS30	.tsafavs.deltaprod/ .tsafavs.applyalt	

Using the PSU Procedure

Table 2-1. Component Name and Product Identifier (continued)

<i>compname</i>	<i>prodid and default appid Identifier</i>	DELTA/APPLY Disks or Directories	Required 4K Blocks (see <i>prodid</i> MEMO)
Note: Use the <i>compname</i> 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.

- As a precaution, do a complete backup of your system. See the *z/VM: Guide for Automated Installation and Service* for instructions on storing a backup copy of the z/VM system to tape.
- Set up the correct virtual machine environment and IPL.

set machine xa

System reset.

System = XA

define storage nnM

STORAGE CLEARED - SYSTEM RESET

STORAGE = nnM

ipl 190 clear

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

Define the storage to at least 256MB.

Defining storage causes a system reset.

z/VM V5.3.0 mm/dd/yy hh:mm

ENTER

Press ENTER to complete the IPL.

- If necessary, access the required disks for the service procedure.

access vdev a

access 5e5 b

access 51d d

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

Access the VMSES/E installation and service tools disk.

Access the software inventory disk R/W.

- 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 *z/VM: VMSES/E Introduction and Reference*.

The variable *ppfname* is the name of your PPF override file. If you are using the override PPF supplied by IBM for z/VM, the *ppfname* is **zvm**. 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.

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 zvm *compname* apply (setup

compname is the component name. See Table 2-1 on page 2-5 for the values unless you are using an override *compname*.

Note

The following tasks, Prepare to Receive a Component from the RSU, Invoke the VMFPSU Command, 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 zvm *compname*

Invoke the VMFPSU Command

Before receiving the service, invoke the VMFPSU command to gather data relating to the RSU and your current system.

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

vmfpsu zvm *compname*

2. Review the beginning of the *appid* PSUPLAN file for the sections that show the VMFPSUxxxx 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.

An example 5VMCMS30 PSUPLAN file for CMS could appear as:

```

*****
****          PPFNAME: ZVM          COMPNAME: CMS
*****
****          PRODID: 5VMCMS30%CMS    Service Level: nnn-0701
*****
****          Date: mm/dd/yy         Time: hh:mm:ss
*****
VMFPSU1071I There are 10 PTFs on the Recommended
              Service Upgrade for PRODID 5VMCMS30%CMS
              that are not currently applied.
VMFPSU1072I There are 2 PTFs currently applied to
              PRODID 5VMCMS30%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 5VMCMS30%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 5VMCMS30%CMS
*****
              PTF.APAR          PTF.APAR          PTF.APAR          PTF.APAR
UM88889.PM88889  UM99997.PM99997
*****
****          PTFs EXCLUDED FOR PRODID 5VMCMS30%CMS
*****
NONE
*****
****          LOCALMODS TO REPROCESS: 5VMCMS30 VVTLCL ON DISK 3C4(E)
*****
PART - ABCDEF TXT
      PTF - UM12345.PM12345
      MOD - LCL0001.UPDT0001
*****
****          LOCALMODS TO REPROCESS: 5VMCMS30 VVTLTST ON DISK 3C4(E)
*****
PART - IJKLM EXC
      PTF - UM11111.PM11111  UM22222.PM22222
      MOD - LCL0002

```

Figure 2-3. Example 5VMCMS30 PSUPLAN

In this example, there are zero PTFs to be excluded (message VMFPSU1076I). PTFs cannot be excluded from the RSU. If a PTF must not be installed, either do not install the RSU or remove the PTF using the VMFREM command after installing the RSU. Message VMFPSU1071I indicates there are 10 PTFs to apply. Message VMFPSU1072I indicates there are 2 PTFs currently applied that will need to be reapplied.

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

Receive a Component from the RSU

Note

Receive the component from the RSU.

1. Receive the Preapplied, Prebuilt Service.

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

If tape:

```
vmfins install ppf zvm compname (nomemo nolink override no
```

If envelope:

```
vmfins install ppf zvm compname (nomemo nolink override no env rptfnum
```

Use `env rptfnum` if the RSU was received as an envelope file.

2. 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, use online HELP or see the appropriate message book.

```
vmfview install
```

Process Additional Service

Note

Apply the component that was received from the RSU.

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.

1. Apply additional service.

```
vmfapply ppf zvm 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.

Using the PSU Procedure

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, use online HELP or see the appropriate message book.

vmfview apply

3. 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.

Also in the figure, if PTF UM12345 is applied (because of APAR PM12345)

```
*****  
****  LOCALMODS TO REPROCESS: 5VMCMS30 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

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 7-8. The rebuild will be done in the next step.

4. If message VMFPSU1078I appears in the PSUPLAN file, a select data file has been updated to force the rebuild of all local service. The necessary rebuilds will occur automatically during Build processing.

Build the New Service Level

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

The build procedure may require that you rebuild a segment or the CMS 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.

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	See . . .
VMSES/E	“Prepare to Build” on page 3-9
REXX/VM	“Prepare to Build” on page 3-16
Language Environment	“Prepare to Build” on page 3-21
CMS	“Prepare to Build” on page 3-28
CP	“Prepare to Build” on page 3-37
GCS	“Prepare to Build” on page 3-45
Dump Viewing Facility	“Prepare to Build” on page 3-52
TSAF	“Prepare to Build” on page 3-57
AVS	“Prepare to Build” on page 3-64

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 2-7.

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 12. Place Your Serviced Components into Production” on page 3-67. 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”).

Place Product Service into Production

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.

After all components have been serviced and tested, follow the instructions in section “Step 12. Place Your Serviced Components into Production” on page 3-67 to rebuild saved segments and copy test build disks to production build disks.

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 z/VM. Corrective service is shipped on a corrective service (COR) tape.

Determine the RSU Service Level Before Ordering COR Service

Use the VMFSIM command to determine the current RSU service level. In the example, 0501 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 ZVM PPF file enter the following VMFSIM command.

```
vmfsim query vm sysrecs tdata :ppf zvm :stat
VMFSIP2480I RESULTS FOR
      TDATA :PPF ZVM :STAT
:PPF ZVM VMSES
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM REXX
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM LE
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM CMS
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM CP
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM GCS
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM DV
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM TSAF
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
:PPF ZVM AVS
      :STAT RECEIVED.dd/mm/yy.hh:mm:ss.MAINT.RSU-0701
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 0701 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 z/VM, you must:

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

Using the COR Service Procedure

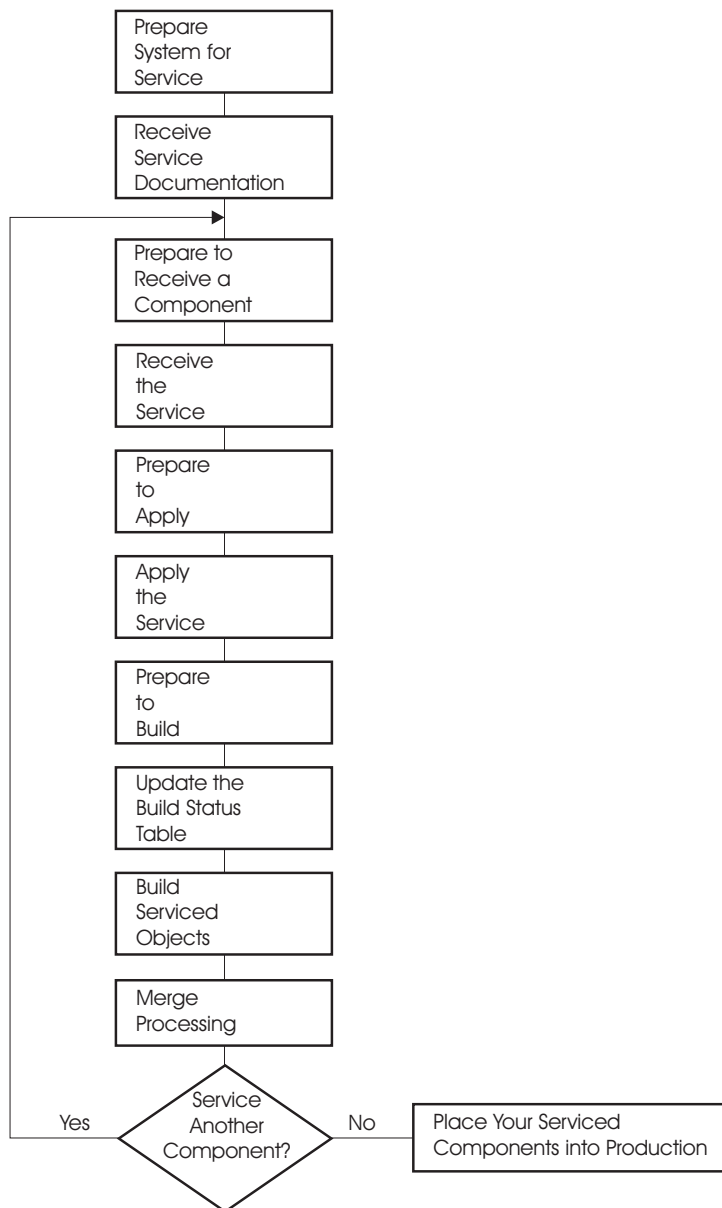


Figure 3-1. Service Application Flowchart using COR

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 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. Copy build disks from test to production.
3. Place the CP nucleus into production.
4. 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 z/VM 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 z/VM components should be serviced in the following order:

1. VMSES/E
2. REXX/VM
3. Language Environment
4. CMS
5. CP
6. GCS
7. Dump Viewing Facility
8. TSAF
9. 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.

Using the COR Service Procedure

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

The COR Service Procedure

The COR service procedure follows. **Follow the steps in order** starting with “Step 1. Prepare the System for Service” on page 3-5.

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 *z/VM: VMSES/E Introduction and Reference*.

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

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

```
set machine xa
System reset.
System = XA
define storage nnM
STORAGE CLEARED - SYSTEM RESET
STORAGE =      nnM
ipl 190 clear
```

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

Define the storage to at least 256 MB.

Defining storage causes a system reset.

z/VM V5.3.0 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
access 5e5 b
access 51d d
release c
```

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

Access the VMSES/E installation and service tools disk.

Access the software inventory disk as R/W.

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 *z/VM: 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 electronically and the documentation envelope file 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, use online HELP or see the appropriate message book.

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

Notes

- If you do not have VMSES/E service on the tape, go to “Step 4. Service REXX/VM” on page 3-13.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMSES30 \$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 **zvm**.
- The *compname* is **vmstes** or **vmstessfs**. If you installed VMSES/E in the shared file system (SFS), use **vmstessfs** as the *compname* instead of **vmstes**. Moving VMSES/E from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is {**vmstes|vmstessfs**}. Enter **vmstes** or **vmstessfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** {**vmstes|vmstessfs**}

Step 3. Service VMSES/E

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {vm ses|vm sessfs}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {vm ses|vm sessfs}
```

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

```
vmfsetup zvm {vm ses|vm sessfs}
```

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 because 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 to this service level.

```
vmfmrdsk zvm {vm ses|vm sessfs} apply
```

- 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

- Apply the service.

```
vmfapply ppf zvm {vm ses|vm sessfs}
```

- 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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
==== 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: ZVM      COMPNAME: VMSES      APPID: 5VMSES30 ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: VMSES/E Introduction and Reference*.

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

```
vmfppf zvm {vm ses|vm sessfs}
```

Step 3. Service VMSES/E

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

```
vmfsetup zvm {vm ses|vm sessfs}
```

Update the Build Status Table

1. Update the Build Status Table with serviced parts.

```
vmfbld ppf zvm {vm ses|vm sessfs} (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 you get the following prompts. You must answer these prompts and enter the following commands. Otherwise, continue to substep 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
```

```
VMFBLD2185R 5VMSES30 $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 5VMSES30 $PPF on 191 (A) from level $PFnnnnn
```

```
vmfppf zvm {vm ses|vm sessfs}
```

```
copy 5VMSES30 $ppf a = = fm-51d (olddate replace  
erase 5VMSES30 $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 zvm {vm ses|vm sessfs}
```

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {vm ses|vm sessfs} (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 “Merge Processing.”

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

```
vmfbld ppf zvm {vm ses|vm sessfs} (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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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
```

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 because 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 to this service level.

```
vmfmrdsk zvm {vm ses|vm sessfs} 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, use online HELP or see the appropriate message book.

Step 3. Service VMSES/E

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 • and this is the last component that you are servicing, then go to	• the next Step. • "Step 12. Place Your Serviced Components into Production" on page 3-67 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 2-11 to receive service for the next component.

Step 4. Service REXX/VM

Notes

- If you do not have REXX/VM service on the tape, go to “Step 5. Service Language Environment” on page 3-19.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMREX30 \$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 **zvm**.
- The *compname* is **rexx** or **rexxsfs**. If you installed REXX/VM in the shared file system (SFS), use **rexxsfs** as the *compname* instead of **rexx**. Moving REXX/VM from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is {**rexx|rexxsfs**}. Enter **rexx** or **rexxsfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** {**rexx|rexxsfs**}

Step 4. Service REXX/VM

Receive the Service

Notes

- If you are receiving service from a disk envelope file, you should use “Receive the Service” on page A-2 to receive service and then return to the next section “Prepare to Apply Service” in this Step. Otherwise, use this procedure to receive service from tape.
- 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 zvm {rex|rexsf}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {rex|rexsf}
```

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

```
vmfsetup zvm {rex|rexsf}
```

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

because 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 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.

```
vmfmrdsd zvm {rex|rexsf} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {rex|rexsf}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
==== 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: ZVM COMPNAME: REXX APPID: 5VMREX30 ****
*****
**** 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 = xxxxxxxx.xxxxxxxx
```

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.

Step 4. Service REXX/VM

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: 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 zvm {rex|rexsf}`

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

`vmfsetup zvm {rex|rexsf}`

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

`vmfbld ppf zvm {rex|rexsf} (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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 on page 3-17 to enter the VMFVIEW command.

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5VMREX30 \$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 5VMREX30 \$PPF on 191 (A) from level \$PFnnnnn

`vmfppf zvm {rex|rexsf}`

`copy 5VMREX30 $ppf a = fm-51d (olddate replace`

`erase 5VMREX30 $ppf a`

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

```
vmfsetup zvm {rex|rexsf}
```

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

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {rex|rexsf} (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 4-4. Then return to “Merge Processing” on page 3-18.

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

```
vmfbld ppf zvm {rex|rexsf} (serviced
```

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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 4-19 and “IPL the CMS Nucleus” on page 4-19. For information on rebuilding and IPLing the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-30 and “IPL the GCS Nucleus” on page 4-31. Then return to “Merge Processing” on page 3-18.

Step 4. Service REXX/VM

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 because 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 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 zvm {rex|rexsf} 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, use online HELP or see the appropriate message book.

```
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 • and this is the last component that you are servicing, then go to	• the next Step. • “Step 12. Place Your Serviced Components into Production” on page 3-67 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 2-11 to receive service for the next component.

Step 5. Service Language Environment

Notes

- If you do not have Language Environment service on the tape, go to “Step 6. Service CMS” on page 3-25.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 4VMVMQ40 \$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 **zvm**.
- The *compname* is **le** or **lesfs**. If you installed Language Environment in the shared file system (SFS), use **lesfs** instead of **le**. Moving Language Environment from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is **{le|lesfs}**. Enter **le** or **lesfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** {**le|lesfs**}

Step 5. Service Language Environment

Receive the Service

Notes

- 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.
- 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 zvm {1e|1esfs}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {1e|1esfs}
```

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

```
vmfsetup zvm {1e|1esfs}
```

4. Enter the VMFMRDSK command to clear the Language Environment 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.

Step 5. Service Language Environment

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 because 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 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.

```
vmfmrdsd zvm {le|lesfs} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {le|lesfs}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
====> 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: ZVM      COMPNAME: LE      APPID: 4VMVMQ40      ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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 Language Environment, 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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF to change the owner of the 51D disk from MAINT to your user ID. For

Step 5. Service Language Environment

examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *z/VM: 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 zvm {1e|lesfs}
```

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

```
vmfsetup zvm {1e|lesfs}
```

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

```
vmfbld ppf zvm {1e|lesfs} (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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 on page 3-23 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
```

```
VMFBLD2185R 4VMVMQ40 $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 4VMVMQ40 $PPF on 191 (A) from level $PFnnnnn
```

```
vmfppf zvm {1e|lesfs}
```

```
copy 4VMVMQ40 $ppf a = = fm-51d (olddate replace  
erase 4VMVMQ40 $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 zvm {1e|lesfs}
```

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {1e|lesfs} (status  
1
```

Step 5. Service Language Environment

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 Language Environment objects that were serviced. If you prefer to build Language Environment objects individually, then use “Detailed Build Steps: Language Environment” on page 4-6. Then return to “Merge Processing.”

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

```
vmfbld ppf zvm {le|lesfs} (serviced
```

Notes:

- a. If you received the following message for objects ILBONBL, ILBONTR, ILBOREC, ILBORNT, ILBOSND, ILBOSNT and ILBOSSN, you can disregard it:

DMSLKD004W Warning messages issued

- b. 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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

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.

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 because 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 to this service level.

```
vmfmrdsk zvm {le|lesfs} apply (setup
```

Step 5. Service Language Environment

2. Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, use online HELP or see the appropriate message book.

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 • and this is the last component that you are servicing, then go to	• the next Step. • "Step 12. Place Your Serviced Components into Production" on page 3-67 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 2-11 to receive service for the next component.

Step 6. Service CMS

Notes

- 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 4-19. 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 7. Service CP” on page 3-34.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMCMS30 \$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 **zvm**.
- The *compname* is **cms** or **cmssfs**. If you installed CMS in the shared file system (SFS), use **cmssfs** as the *compname* instead of **cms**. Moving CMS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is {**cms|cmssfs**}. Enter **cms** or **cmssfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** {**cms|cmssfs**}

Step 6. Service CMS

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {cms|cmssfs}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {cms|cmssfs}
```

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

```
vmfsetup zvm {cms|cmssfs}
```

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

because 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 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.

```
vmfmrchk zvm {cms|cmssfs} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {cms|cmssfs}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
==== 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: ZVM      COMPNAME: CMS      APPID: 5VMCMS30 ****
*****
****          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 = xxxxxxxx.xxxxxxxx
```

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.

Step 6. Service CMS

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 *z/VM: 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 zvm {cms|cmssfs}`

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

`vmfsetup zvm {cms|cmssfs}`

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:

`vmfblld ppf zvm {cms|cmssfs} (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 substep 3 on page 3-29 to enter the VMFVIEW command.

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5VMCMS30 \$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 5VMCMS30 $PPF on 191 (A) from level $PFnnnnn
vmfppf zvm {cms|cmssfs}
```

```
copy 5VMCMS30 $ppf a = = fm-51d (olddate replace
erase 5VMCMS30 $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 zvm {cms|cmssfs}
```

Enter VMFSETUP to ensure correct access order.

```
vmfbld ppf zvm {cms|cmssfs} (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 4-12. Then return to “Test the New Level of CMS” on page 3-32.

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 5VMCMS30 srvblds * tdata :object defnuc :stat
VMFSIP2480I RESULTS FOR
          TDATA :OBJECT DEFNUC
:BLDLIST DMSOM
:OBJECT DEFNUC
:STAT SERVICED.mm/dd/yy.hh:mm:ss.MAINT
```

Step 6. Service CMS

If the STAT field indicates that DEFNUC has been serviced see “Update the Macro Libraries (MACLIBs)” on page 4-14 for instructions to update the DMSOM MACLIB.

```
vmf-assemble dmsngp zvm {cms|cmssfs} ($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 *z/VM: CMS Planning and Administration*.

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.

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

```
vmfbld ppf zvm {cms|cmssfs} (serviced
```

Notes:

- a. If you receive the following message:

DMSLI0201W 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:

DMSLI0201W 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 globalled.

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 12. Place Your Serviced Components into Production” on page 3-67.

- e. If, while building the DMSCUT MODULE or the DMSDFT MODULE, you get a return code of 4 with this message:

DMSLI0201W The following names are undefined: DMSUSRX1 DMSUSRX2

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

4. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

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 3-32.

```
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 *z/VM: CMS 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 12. Place Your Serviced Components into Production” on page 3-67.

```
z/VM V5.3.0 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).

Step 6. Service CMS

ENTER

Press ENTER to complete the IPL.

4. Save the CMS load map.

```
access 493 fm-493
spool prt nohold
close prt
RDR FILE 0026 SENT FROM MAINT PRT
  WAS 0026 RECS 8270 CPY 001 A NOHOLD NOKEEP
receive spoolid cmsnuc map fm-493 (replace

vmfcopy cmsnuc map fm-493 = = = (prodid 5VMCMS30%cms replace pack
```

fm-493 is any free file mode.

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

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.

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 4-19 and “IPL the CMS Nucleus” on page 4-19. Then return to “Test the New Level of CMS.”

```
xedit cmsnuc map
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found
====> all/undefin/
DMSXDC546E Target not found
====> qquit
```

This response means that no unresolved symbols were found.

This response means that no undefined symbols were found.

6. Redefine the CMS disks back to their original values:

```
define 190 490
define 590 190
ipl 190 clear
z/VM V5.3.0 mm/dd/yy hh:mm
ENTER
```

Press ENTER to complete the IPL.

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 “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 because 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 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 zvm {cms|cmssfs} 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, use online HELP or see the appropriate message book.

```
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 • and this is the last component that you are servicing, then go to	• the next Step. • “Step 12. Place Your Serviced Components into Production” on page 3-67 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 2-11 to receive service for the next component.

Step 7. Service CP

Notes

- If you do not have CP service on the tape, go to “Step 8. Service GCS” on page 3-42.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMCPR30 \$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 **zvm**.
- The *compname* is **cp** or **cpsfs**. If you installed CP in the shared file system (SFS), use **cpsfs** as the *compname* instead of **cp**. Moving CP from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is {**cp|cpsfs**}. Enter **cp** or **cpsfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** {**cp|cpsfs**}

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {cp|cpsfs}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {cp|cpsfs}
```

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

```
vmfsetup zvm {cp|cpsfs}
```

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

Step 7. Service CP

because 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 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.

```
vmfmrdsd zvm {cp|cpsfs} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {cp|cpsfs}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
====> 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: ZVM      COMPNAME: CP      APPID: 5VMCPR30      ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: 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 zvm {cp|cpsfs}
```

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

```
vmfsetup zvm {cp|cpsfs}
```

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

```
vmfbld ppf zvm {cp|cpsfs} (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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 on page 3-38 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
```

```
VMFBLD2185R 5VMCPR30 $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 5VMCPR30 $PPF on 191 (A) from level $PFnnnnn
```

```
vmfppf zvm {cp|cpsfs}
```

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

Step 7. Service CP

```
copy 5VMCPR30 $ppf a = fm-51d (olddate replace
erase 5VMCPR30 $ppf a
```

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

```
vmfsetup zvm {cp|cpsfs}
vmfbld ppf zvm {cp|cpsfs} (status
1
```

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. Enter VMFSETUP to ensure correct access order. 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 4-22. Then return to “Merge Processing” on page 3-40.

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

```
vmfbld ppf zvm {cp|cpsfs} (serviced
```

Notes:

- a. If you receive the following:

```
CSECT'S WITH SIZE GREATER THAN CONDITIONAL PAGE BOUNDARY
@MAPSTRT
```

disregard it. For more information, see *z/VM: CP Commands and Utilities Reference*.

- b. 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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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
z/VM V5.3.0 mm/dd/yy hh:mm
ENTER
```

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

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 maps, see “Rebuild the CP Nucleus” on page 4-25 and Verify the CP Load Map. Then continue with “Merge Processing” on page 3-40.

```
xedit cpload map fm-493
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found
====> all/undefin/
DMSXDC546E Target not found
====> qquit
```

fm-493 is the file mode of the target disk which is where the CPLOAD MAP was placed when it was created. This response means that no unresolved symbols were found. This response means that no undefined symbols were found.

Create a Back-Up Copy of the Primary Parm Disk

Make a copy of the current parm disk.

1. Link and access the CF3 parm disk in write mode.

```
cprelease c
link maint cf3 cf3 m
access cf3 fm3
```

fm3 is any free filemode

2. Use ACCESSM0 command to allow files with a file mode of '0' to be copied. Then copy all files from the CF1 parm disk to the CF3 parm disk.

```
access 493 r
accessm0 on
access cf1 fm1 (mode0
copyfile * * fm1 = = fm3 (olddate replace
```

fm1 is any free filemode

Step 7. Service CP

Create the New CP Nucleus on the Secondary Parm Disk

Make a test parm disk, from the current parm disk and the new CP nucleus, on the Secondary Parm disk.

1. Link and access the CF2 parm disk in write mode.

```
cprelease b
link maint cf2 cf2 m
access cf2 fm2 (erase
```

fm2 is any free filemode

2. Copy all files from the CF1 parm disk to the CF2 parm disk.

```
copyfile * * fm1 = = fm2 (olddate replace
```

3. Rename the current CPLOAD MODULE and copy the new CPLOAD MODULE.

```
erase cplold module fm2
rename cpload module fm2 cplold = =
copyfile cpload module r = = fm2 (olddate replace
```

Erase the backup CP nucleus

Copy the new CP nucleus to the parm disk.

4. Reset your environment.

```
release fm3 (detach
cpaccess maint cf3 c rr
release fm2 (detach
cpaccess maint cf2 b rr
release fm1
accessm0 off
release r
```

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 because 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 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 zvm {cp|cpsfs} 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, use online HELP or see the appropriate message book.

```
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 • and this is the last component that you are servicing, then go to	• the next Step. • "Step 12. Place Your Serviced Components into Production" on page 3-67 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 2-11 to receive service for the next component.

Step 8. Service GCS

Notes

- 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 4-30 and then “IPL the GCS Nucleus” on page 4-31. If you do not have GCS and REXX/VM service, skip this step and go to “Step 9. Service the Dump Viewing Facility” on page 3-49.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMGCS30 \$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 **zvm**.
- The *compname* is **gcs** or **gcssf**s. If you installed GCS in the shared file system (SFS), use **gcssf**s instead of **gcs**. Moving GCS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is **{gcs|gcssf**s}. Enter **gcs** or **gcssf**s. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm** **{gcs|gcssf**s}

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {gcs|gcssf}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {gcs|gcssf}
```

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

```
vmfsetup zvm {gcs|gcssf}
```

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

Step 8. Service GCS

because 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 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.

```
vmfmrdsd zvm {gcs|gcssf} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {gcs|gcssf}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
====> 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: ZVM      COMPNAME: GCS      APPID: 5VMGCS30      ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: 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 zvm {gcs|gcssf}
```

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

```
vmfsetup zvm {gcs|gcssf}
```

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 *z/VM: Group Control System*.

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 zvm {gcs|gcssf} (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 substep 3 on page 3-46 to enter the VMFVIEW command.

Step 8. Service GCS

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5VMGCS30 \$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 5VMGCS30 \$PPF on 191 (A) from level \$PFnnnnn

vmfppf zvm {gcs|gcssf}

Enter the VMFPPF command to ensure you have the appropriate level of PPF. *fm-51d* is the file mode of the software inventory disk.

copy 5VMGCS30 \$ppf a = = fm-51d (olddate replace

erase 5VMGCS30 \$ppf a

vmfsetup zvm {gcs|gcssf}

Enter the VMFSETUP command to ensure correct access order.

vmfbld ppf zvm {gcs|gcssf} (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 any new GCS nucleus build list(s) that you added will need to be built. After issuing the VMFBLD command during step 1 in “Build Serviced Objects” issue the following VMFBLD command:

vmfbld ppf zvm {gcs|gcssf} newbldlist (all

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 use “Detailed Build Steps: GCS” on page 4-28. Then return to “Merge Processing” on page 3-48.

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

vmfbld ppf zvm {gcs|gcssf} (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 12. Place Your Serviced Components into Production” on page 3-67.

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, use online HELP or see the appropriate message book.

```
vmfview build
```

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 skip this section and continue with “Merge Processing” on page 3-48.

```
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
```

```
NAMED SAVED SYSTEM (NSS) GCS
WAS SUCCESSFULLY SAVED IN FILEID 0100
RDR FILE spoolid SENT FROM MAINT
```

Save the nucleus in your saved segment.

This message indicates that your nucleus was successfully saved.

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

Restore your CMS environment.

```
ipl 190 clear
```

```
z/VM V5.3.0 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. Save the load map on the test system tools disk (493).

```
access 493 fm-493
```

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

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

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 5VMGCS30%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.

Step 8. Service GCS

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 4-30 and “IPL the GCS Nucleus” on page 4-31. Then return to “Merge Processing.”

```
xedit gcsnuc map
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found

====> all/undefin/
DMSXDC546E Target not found
====> qquit
release fm-493
```

This response means that no unresolved symbols were found.

This response means that no undefined symbols were found.

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

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 because 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 to this service level.

```
vmfmrdsk zvm {gcs|gcssf} 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, use online HELP or see the appropriate message book.

```
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 • and this is the last component that you are servicing, then go to	• the next Step. • “Step 12. Place Your Serviced Components into Production” on page 3-67 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 2-11 to receive service for the next component.

Step 9. Service the Dump Viewing Facility

Note

- If you do not have the Dump Viewing Facility service on the tape, go to “Step 10. Service TSAF” on page 3-55.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component 5VMDVF30 \$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 **zvm**.
- The *compname* is **dv** or **dvsfs**. If you installed the Dump Viewing Facility in the shared file system (SFS), use **dvsfs** as the *compname* instead of **dv**. Moving the Dump Viewing Facility from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is **{dv|dvsfs}**. Enter **dv** or **dvsfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
2. Make sure 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 **zvm** {dv|dvsfs}

Step 9. Service the Dump Viewing Facility

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {dv|dvsfs}
```

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, use online HELP or see the appropriate message book.

```
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 zvm {dv|dvsfs}
```

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

```
vmfsetup zvm {dv|dvsfs}
```

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.

Step 9. Service the Dump Viewing Facility

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 because 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 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.

```
vmfmrdsd zvm {dv|dvsfs} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {dv|dvsfs}
```

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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
====> 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: ZVM      COMPNAME: DV      APPID: 5VMDVF30      ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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.

Step 9. Service the Dump Viewing Facility

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: 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 zvm {dv|dvsfs}`

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

`vmfsetup zvm {dv|dvsfs}`

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

`vmfbld ppf zvm {dv|dvsfs} (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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 on page 3-53 to enter the VMFVIEW command.

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5VMDVF30 \$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 5VMDVF30 \$PPF on 191 (A) from level \$PFnnnnn

`vmfppf zvm {dv|dvsfs}`

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

`copy 5VMDVF30 $ppf a = fm-51d (olddate replace
erase 5VMDVF30 $ppf a`

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

Step 9. Service the Dump Viewing Facility

```
vmfsetup zvm {dv|dvsfs}
```

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {dv|dvsfs} (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 4-33. Then return to “Merge Processing.”

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

```
vmfbld ppf zvm {dv|dvsfs} (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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

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 because 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 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 zvm {dv|dvsfs} apply (setup
```

Step 9. Service the Dump Viewing Facility

2. Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, use online HELP or see the appropriate message book.

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 • and this is the last component that you are servicing, then go to	• the next Step. • "Step 12. Place Your Serviced Components into Production" on page 3-67 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 2-11 to receive service for the next component.

Step 10. Service TSAF

Notes

- If you do not have TSAF service on the tape, go to “Step 11. Service AVS” on page 3-61.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMTSA30 \$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 **zvm**.
- The *compname* is **tsaf** or **tsafsfs**. If you installed TSAF in the shared file system (SFS), use **tsafsfs** as the *compname* instead of **tsaf**. Moving TSAF from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is **{tsaf|tsafsfs}**. Enter **tsaf** or **tsafsfs**. Do not enter the “|”.

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
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 **zvm {tsaf|tsafsfs}**

Step 10. Service TSAF

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {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, use online HELP or see the appropriate message book.

```
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 zvm {tsaf|tsafsfs}
```

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

```
vmfsetup zvm {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

because 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 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.

```
vmfmrdsd zvm {tsaf|tsafsfs} 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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
====> 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: ZVM      COMPNAME: TSAF      APPID: 5VMTSA30      ****
*****
****      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 = xxxxxxxx.xxxxxxxx
```

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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF to change the owner of the 51D disk from MAINT to your user ID. For

Step 10. Service TSAF

examples of changing these files, see the chapter on changing the Software Inventory to an SFS directory in the *z/VM: 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 zvm {tsaf|tsafsfs}
```

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

```
vmfsetup zvm {tsaf|tsafsfs}
```

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

```
vmfbld ppf zvm {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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 to enter the VMFVIEW command.

```
VMFBLD2185R The following source product parameter files have been serviced:
```

```
VMFBLD2185R 5VMTSA30 $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 5VMTSA30 $PPF on 191 (A) from level $PFnnnnn
```

```
vmfppf zvm {tsaf|tsafsfs}
```

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

```
copy 5VMTSA30 $ppf a = = fm-51d (olddate replace
```

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

```
erase 5VMTSA30 $ppf a
```

```
vmfsetup zvm {tsaf|tsafsfs}
```

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {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 4-36. Then return to “Merge Processing.”

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

```
vmfbld ppf zvm {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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

vmfview build

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 because 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 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 zvm {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, use online HELP or see the appropriate message book.

vmfview merge

Step 10. Service TSAF

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 • and this is the last component that you are servicing, then go to	• the next Step. • “Step 12. Place Your Serviced Components into Production” on page 3-67 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 2-11 to receive service for the next component.

Step 11. Service AVS

Notes

- If you do not have AVS service on the tape, you have finished installing service for z/VM components.
- If you are using the PPF override file supplied by IBM for z/VM, the *ppfname* is **zvm**. The PPF override file contains overrides to the base PPF file for this component, 5VMAVS30 \$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 **zvm**.
- The *compname* is **avs** or **avssfs**. If you installed AVS in the shared file system (SFS), use **avssfs** as the *compname* instead of **avs**. Moving AVS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.
- In the following procedures, the notation for the component name is **{avslavssfs}**. Enter **avs** or **avssfs**. Do not enter the "I".

Prepare to Receive Service

1. Unless you are receiving service from a disk envelope file, make sure that a tape drive is attached to your virtual machine and that the COR tape is mounted and ready.
2. Make sure 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 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 **zvm** {**avs|avssfs**}

Step 11. Service AVS

Receive the Service

Notes

- If you are receiving service from a 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.
- 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 zvm {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, use online HELP or see the appropriate message book.

```
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 zvm {avs|avssfs}
```

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

```
vmfsetup zvm {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

because 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 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.

```
vmfmrdsd zvm {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 *z/VM: VMSES/E Introduction and Reference*.

Apply the Service

1. Apply the service.

```
vmfapply ppf zvm {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, use online HELP or see the appropriate message book.

```
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 rework local service.

```
==== 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: ZVM      COMPNAME: AVS      APPID: 5VMAVS30      ****
*****
****      Date: mm/dd/yy      Time: hh:mm:ss      ****
*****
CK:VMFAPP2120W Part fn ftabbrev has service at level VVTLCCL which is a
CK:      higher level than VVTVM. The higher-level service may need
CK:      to be reworked or removed.
CK:VMFAPP2121I VVTLCCL = xxxxxxxx.xxxxxxxx
```

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.

Step 11. Service AVS

Prepare to Build

Note

If you use a user ID other than MAINT to apply service, you must update the file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: 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 zvm {avs|avssfs}`

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

`vmfsetup zvm {avs|avssfs}`

Update the Build Status Table

1. Update the Build Status Table with serviced parts:

`vmfbld ppf zvm {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 you get the following prompts. You must answer the prompts and enter the following commands. Otherwise, continue to substep 3 on page 3-65 to enter the VMFVIEW command.

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5VMAVS30 \$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 5VMAVS30 \$PPF on 191 (A) from level \$PFnnnnn

`vmfppf zvm {avs|avssfs}`

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

`copy 5VMAVS30 $ppf a = fm-51d (olddate replace`

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

`erase 5VMAVS30 $ppf a`

`vmfsetup zvm {avs|avssfs}`

Enter the VMFSETUP command to ensure correct access order.

```
vmfbld ppf zvm {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 4-38. Then return to “Merge Processing.”

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

```
vmfbld ppf zvm {avs|avssfs} (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 12. Place Your Serviced Components into Production” on page 3-67.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

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 because 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 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 zvm {avs|avssfs} apply (setup
```

Step 11. Service AVS

2. Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific merge messages, use online HELP or see the appropriate message book.

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 12. Place Your Serviced Components into Production" on page 3-67 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 2-11 to receive service from the PSP bucket COR tape.

Step 12. Place Your Serviced Components into Production

This step assumes you have performed any testing you find necessary on the service you have received and are ready to put it into production. It also assumes you have serviced all components for this service session. In this procedure you will:

- Rebuild the saved segments.
- 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.
- 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 *z/VM: Saved Segments Planning and Administration*.

1. The CMS saved segments are defined using the ZVM 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 *z/VM: 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 file VMSESE PROFILE and the :DCL section of the file SEGBLD PPF 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 *z/VM: VMSES/E Introduction and Reference*.

3. Prepare to build the saved segments.

Step 12. Place Components into Production

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
ipl 190 clear parm nosprof instseg no
```

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 CMS without executing the system profile (SYSPROF EXEC) (which, for example, loads the CMSPIPES saved segment) and without loading the installation saved segment (CMSINST).

z/VM V5.3.0 mm/dd/yy hh:mm

**** DO NOT press ENTER! ****

access (nosprof

Bypass the execution of the PROFILE EXEC.

4. Establish the minidisk access order and update the build status table.

```
access 5e6 b
access 51d d
vmfppf segbld esasegs
vmfbld ppf segbld esasegs (status
vmfview build
```

Access the test VMSES/E installation and service tools disk.

Access the software inventory disk as R/W.

segbld is a PPF and **esasegs** is a special component name for building segments.

Enter this command to determine which segments need to be built because they were serviced.

View the build message log.

Note

If you would like to build the individual segments one at a time skip this note and continue with 5 on page 3-69.

Otherwise, if you want to build all serviced CMS segments enter the following command and then go to 20 on page 3-70.

vmfbld list segbld esasegs zvmsegs 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
```

Step 12. Place Components into Production

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 NLSUCENG logical saved segment.

```
vmfbld ppf segbld esasegs segblist nlsuceng (serviced
```

15. Build the NLSGER logical saved segment.

```
vmfbld ppf segbld esasegs segblist nlsger (serviced
```

16. Build the NLSKANJI logical saved segment.

```
vmfbld ppf segbld esasegs segblist nlskanji (serviced
```

17. Build the NLSAMENG logical saved segment.

```
vmfbld ppf segbld esasegs segblist nlsameng (serviced
```

18. Build the SCEE logical saved segment.

```
vmfbld ppf segbld esasegs segblist scee (serviced
```

19. Build the SCEEX logical saved segment.

```
vmfbld ppf segbld esasegs segblist sceex (serviced
```

20. If necessary, correct any problems before going on. You might see other build requirements not relating to z/VM that will need to be resolved.

```
vmfview build
```

21. 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.

22. If you received the following message:

```
VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and  
must be moved to the S disk.
```

Step 12. Place Components into Production

from any of the preceding builds 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 3-72.

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
```

Move Files into the Byte File System

The following section provides information on moving Shell and Utilities files into the byte file system after they have been serviced. You only need to move these files if you have installed Shell and Utilities 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 zvm {cmsprod|cmsprodsfs}
vmfsetup zvm {cmsprod|cmsprodsfs}
vmfbld ppf zvm {cmsprod|cmsprodsfs} (status
vmfview build
```

cmsprod or **cmsprodsfs** is a special component name used for CMS production steps. If you installed CMS in the shared file system (SFS), use **cmsprodsfs** instead of **cmsprod**.

2. Move the Shell and Utilities files to the byte file system.

```
vmfbld ppf zvm {cmsprod|cmsprodsfs} gsublstg (serviced
```

3. If necessary, correct any problems before going on.

```
vmfview build
```

Step 12. Place Components into Production

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 VMFCOPY command to copy the Language Environment files from the test system product disk (49E) to the production system product disk (19E).

```
access 49e n
access 19e t
vmfcopy * * n = = t2 (prodid 4VMVMQ40%le olddate replace
```

3. 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
```

4. 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:
```

```
sysprint cons
```

The system prompts you to enter a DDR control statement.

Route messages to the console (rather than a printer).

Step 12. Place Components into Production

ENTER:
input 490 dasd mnt490

ENTER:
output 190 dasd mnt190

ENTER:
copy all
END OF COPY
ENTER:

ENTER

END OF JOB

access 190 r
format 190 r (label
DMSFOR605R Enter disk label:
mnt190

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, 3390). **mnt490** is the label of the minidisk. 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 the COPY control statement to copy the entire 490 minidisk.

DDR copies and displays messages about the operation. When the processing is complete, DDR prompts you to enter another control statement.

At the prompt, press ENTER to enter a null line.

The system reports end-of-job status.

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
z/VM V5.3.0 mm/dd/yy hh:mm

ENTER

Build the CMS saved system.

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 z/VM. Then copy the new CPLOAD MODULE, which is your CP nucleus, to the primary parm disk CF1. Enter the following commands.

Step 12. Place Components into Production

query cpdisk

LABEL	USERID	VDEV	MODE	STAT	VOL-ID	RDEV	TYPE	STARTLOC	ENDLOC
MNTCF1	MAINT	0CF1	A	R/O	530RES	rdev	ECKD	nn	nn
MNTCF2	MAINT	0CF2	B	R/O	530RES	rdev2	ECKD	nn	nn
MNTCF3	MAINT	0CF3	C	R/O	530RES	rdev3	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 cf2 t  
erase cplold module u
```

Erase the backup CP nucleus.

```
rename cpload module u cplold module u
```

Note: CPLOLD module is the default name used for the backup copy of the CP nucleus during installation. 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 operator's guide for your hardware 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 *z/VM: CP Planning and Administration*.

If you could not use the CF1 parm disk, repeat this process with the CF2 parm disk.

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 3-1. Component Name

<i>compname</i>	<i>prodid and default appid Identifier</i>
VMSES or VMSESSFS	5VMSES30
REXX or REXXSFS	5VMREX30
LE or LESFS	4VMVMQ40
CMS or CMSSFS	5VMCMS30
CP or CPSFS	5VMCPR30
DV or DVSFS	5VMDVF30
GCS or GCSSFS	5VMGCS30
TSAF or TSAFSFS	5VMTSA30
AVS or AVSSFS	5VMAVS30

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 3-1, then enter the following command.

```
vmfppf zvm compname
```

3. Enter the following command.

```
vmfsetup zvm compname
```

Step 12. Place Components into Production

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 3-1.

```
vmfmrdsd zvm compname apply (levels 2  
vmfmrdsd zvm compname apply (levels 2
```

You need to enter VMFMRDSK twice to ensure that all service levels are merged to the apply production disk. Because 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.

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 z/VM. 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

Prepare to Build

Note

The *compname* is **vmstes** or **vmstessfs**. If you installed VMSES/E in the shared file system (SFS), use **vmstessfs** as the *compname* instead of **vmstes**. Moving VMSES/E from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**vmstes|vmstessfs**}. Enter **vmstes** or **vmstessfs**. Do not enter the “|”.

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

If the build list is:	Then follow this build procedure:
VMFBLHLP VMFBLNON VMFBLSES VMFBLSRC VMFBLSYS VMFBLUCE	“Select and Copy Serviced Files”
VMFMLOAD VMFSLOAD	“Rebuild the Executable MODULEs” on page 4-2
VMFSBHLP	“Process the Segment Build List” on page 4-2

Select and Copy Serviced Files

Table 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

Detailed Build Steps: VMSES/E

Table 4-2. Build lists for VMSES/E select and copy functions (continued)

Build List	Description	Test/Production Disk
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
VMFBLUCE	Build list for the Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-2 that was in the \$VMFBLD \$MSGLOG.

`vmfbld ppf zvm {vm ses|vm sessfs} bldlist (serviced` *bldlist* is the file name of one of the build lists in Table 4-2 on page 4-1.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`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 zvm {vm ses|vm sessfs} 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, use online HELP or see the appropriate message book.

`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 12. 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 zvm {vm ses|vm sessfs} vmfshlp (serviced
```

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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 “Merge Processing” on page 3-11 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: REXX/VM

Prepare to Build

Note

The *compname* is **rexx** or **rexxsfs**. If you installed REXX/VM in the shared file system (SFS), use **rexxsfs** as the *compname* instead of **rexx**. Moving REXX/VM from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**rexx|rexxsfs**}. Enter **rexx** or **rexxsfs**. Do not enter the “|”.

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

If the build list is:	The follow this build procedure:
IXXBLSRC	“Select and Copy Serviced Files”
IXXOM	“Update the Macro Libraries (MACLIBs)” on page 4-5
IXXBLSNUC	“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 zvm {rexx|rexxsfs} ixxbldsrc (serviced
```

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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.”

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 zvm {rex|rexsf} ixom (serviced` `ixom` is the name of the MACLIB from Table 4-7.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`vmfview build`

Rebuild the Nuclei

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 4-19 and “IPL the CMS Nucleus” on page 4-19. For information on rebuilding and IPLing the GCS nucleus, see “Rebuild the GCS Nucleus” on page 4-30 and “IPL the GCS Nucleus” on page 4-31.

You have now finished the detailed build steps for REXX/VM. Go to “Merge Processing” on page 3-18 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: Language Environment

Prepare to Build

Note

The *compname* is **le** or **lesfs**. If you installed Language Environment in the shared file system (SFS), use **lesfs** as the *compname* instead of **le**. Moving Language Environment from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is **{lelesfs}**. Enter **le** or **lesfs**. Do not enter the “l”.

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. Language Environment build lists

If the build list is:	Then follow this build procedure:
CEEBLFPR EDCBL49B EDCBLFPR EDCBLH EDCBLHPE EDCBLHPU IBMBLFPR IGZBLFPR	“Select and Copy Serviced Files” on page 4-7
CEEBLMAC CEEBSAM EDCBLMAC IBMBLSAM IGZBLMAC IGZBSAM	“Update the Macro Libraries (MACLIBs)” on page 4-8
CEEBLKLD CEEBLOBJ CEEBSB2 EDCBLCPP EDCBLKLD EDCBSB2 EDCBLSP1 EDCBLSP2 IBMBLCAL IBMBLLKD IBMBLMTH IGZBLLKD	“Rebuild the Text Libraries” on page 4-8

Table 4-8. Language Environment build lists (continued)

If the build list is:	Then follow this build procedure:
CEEBLRUN EDCBLRUN IBMBLRUN IGZBLLBO IGZBLRUN	"Rebuild the Load Libraries" on page 4-9
CEEBLMOD CEEBLNLS EDCBLMOD EDCBLPMB IBMBLMOD IBMBLNLS IGZBLMOD IGZBLNLS IGZBLPAC	"Rebuild the Executable MODULEs" on page 4-9
CEEBLSGA CEEBSGB EDCSBHLP	"Process the Segment Build Lists" on page 4-10

Select and Copy Serviced Files

Table 4-9 specifies build lists for Language Environment that require the select and copy build functions.

Table 4-9. Build lists for Language Environment select and copy functions

Build List	Description	Test/Production Disk
CEEBLFPR	Build list for full part replacement "CEL" files	49E/19E
EDCBL49B	Build list for full part replacement locale information	49B/49B
EDCBLFPR	Build list for full part replacement "C" files	49E/19E
EDCBLH	Build list for C/VM™ header files	49E/19E
EDCBLHPE	Build list for HELP files	19D/19D
EDCBLHPU	Build list for uppercase HELP files	402/402
IBMBLFPR	Build list for full part replacement "COBOL" files	49E/49E
IGZBLFPR	Build list for full part replacement "PL/I" files	49E/49E

1. Enter the following command for each build list in Table 4-9 that was identified in the \$VMFBLD \$MSGLOG.

```
vmfbld ppf zvm {le|lesfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-9 on page 4-7.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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.”

Table 4-10 contains a list of Language Environment MACLIBs that may need to be rebuilt.

Table 4-10. Build lists for Language Environment MACLIBs

Build Lists	Description	Test/Production Disk
CEEBLMAC	Build list for the SCEEMAC MACLIB	49E/19E
CEEBSAM	Build list for the SCEESAMP MACLIB	49E/19E
EDCBLMAC	Build list for the SCEEMAC MACLIB	49E/19E
IBMBLSAM	Build list for the SCEESAMP MACLIB	49E/49E
IGZBLMAC	Build list for the SCEEMAC MACLIB	49E/49E
IGZBSAM	Build list for the SCEESAMP MACLIB	49E/49E

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 zvm {le|lesfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-10 on page 4-8.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Text Libraries

Table 4-11 describes the build lists used to rebuild the Language Environment TXTLIBs.

Table 4-11. Build lists for Language Environment TXTLIBs

Build Lists	Description	Test/Production Disk
CEEBLKLD	Build files for the SCEELKED TXTLIB	49E/19E
CEEBOBJ	Build files for the SCEEOBJ TXTLIB	49E/19E
CEEBSB2	Build files for the SCEEBND2 TXTLIB	49E/49E
EDCBLCPP	Build files for the SCEECPP TXTLIB	49E/19E
EDCBLLKD	Build files for the SCEELKED TXTLIB	49E/19E
EDCBLB2	Build files for the SCEEBND2 TXTLIB	49E/19E
EDCBLSP1	Build files for the SCEESPC TXTLIB	49E/19E
EDCBLSP2	Build files for the SCEESPCO TXTLIB	49E/19E

Table 4-11. Build lists for Language Environment TXTLIBs (continued)

Build Lists	Description	Test/Production Disk
IBMBLCAL	Build files for the SIBMCALL TXTLIB	49E/49E
IBMBLLKD	Build files for the SCEELKD TXTLIB	49E/49E
IBMBLMTH	Build files for the SIBMMATH TXTLIB	49E/49E
IGZBLLKD	Build files for the SCEELKD TXTLIB	49E/49E

1. For each build list in Table 4-11 on page 4-8 that has been serviced, enter the following command to create the updated level of the TXTLIB associated with it.

```
vmfbl d ppf zvm {le|lesfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-11 on page 4-8.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Load Libraries

Table 4-12 contains the load libraries that may need to be rebuilt.

Table 4-12. Build list for Language Environment load libraries

Build List	Description	Test/Production Disk
CEEBLRUN	Build list for the SCEERUN LOADLIB	49E/19E
EDCBLRUN	Build list for the SCEERUN LOADLIB	49E/19E
IBMBLRUN	Build list for the SCEERUN LOADLIB	49E/49E
IGZBLLBO	Build list for the SCEELBO LOADLIB	49E/49E
IGZBLRUN	Build list for the SCEERUN LOADLIB	49E/49E

1. Enter the VMFBLD command to rebuild the load library for each build list that needs to be rebuilt.

```
vmfbl d ppf zvm {le|lesfs} bldlist (serviced
```

bldlist is a build list from Table 4-12.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Executable MODULES

Table 4-13 describes the build lists used to rebuild executable MODULES.

Table 4-13. Build lists for Language Environment executable MODULES

Build List	Description	Test/Production Disk
CEEBLMOD	Build list for CEL modules	49E/19E

Detailed Build Steps: Language Environment

Table 4-13. Build lists for Language Environment executable MODULEs (continued)

Build List	Description	Test/Production Disk
CEEBLNLS	Build list for NLS modules	49E/19E
EDCBLMOD	Build list for C MODULEs	49E/19E
EDCBLPMB	Build list for C MODULEs built with the binder/loader	49E/19E
IBMBLMOD	Build list for the PL/I modules	49E/49E
IBMBLNLS	Build list for the NLS modules	49E/49E
IGZBLMOD	Build list for the COBOL modules	49E/49E
IGZBLNLS	Build list for the NLS modules	49E/49E
IGZBLPAC	Build list for the COBOL modules	49E/49E

1. For each build list in Table 4-13 on page 4-9 that has MODULEs that have been serviced, enter the VMFBLD command to rebuild the serviced modules.

```
vmfbld ppf zvm {le|lesfs} bldlist (serviced
```

bldlist is a file name of one of the build lists in Table 4-13 on page 4-9.

If you received the following message for objects ILBONBL, ILBONTR, ILBOREC, ILBORNT, ILBOSND, ILBOSNT and ILBOSSN, you can disregard it:

DMSLKD004W Warning messages issued

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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 12. Place Your Serviced Components into Production."

Table 4-14. Build lists for Language Environment segment build lists

Build List	Description
CEEBLSGA	Build list for the SCEEX saved segment
CEEBLSGB	Build list for the SCEE saved segment
EDCSBHLP	Build list for Language Environment files that reside with the HELP segment

1. Enter the following command for each segment build list, if it has been serviced.

```
vmfbld ppf zvm {le|lesfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-14 on page 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, use online HELP or see the appropriate message book.

Note

You have now finished the detailed build steps for Language Environment. Go to “Merge Processing” on page 3-23 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: CMS

Prepare to Build

Note

The *compname* is **cms** or **cmssfs**. If you installed CMS in the shared file system (SFS), use **cmssfs** as the *compname* instead of **cms**. Moving CMS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**cms|cmssfs**}. Enter **cms** or **cmssfs**. Do not enter the “|”.

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 4-15, and then run the corresponding build steps in the order that they appear in the table.

Table 4-15. CMS build lists

If the build list is:	Then follow this build procedure:
DMSBL400 DMSBL490 DMSBL493 DMSBL990 DMSBLASM DMSBLCRX DMSBLDOS DMSBLHLP DMSBLINF DMSBLNUH DMSBLRAS DMSBLSRC GSUBLSYS DMSBLUCE	“Select and Copy Serviced Files” on page 4-13
AMQOM DMSGPI DMSOM FPLGPI FPLOM MVSXA OSMACRO OSMACRO1 OSPSI	“Update the Macro Libraries (MACLIBs)” on page 4-14
CMSSAA DMSBLAEN DMSBLAMT DMSBLBKW DMSBLCEN DMSBLCMT GSUBLLIB	“Rebuild the Text Libraries” on page 4-15

Table 4-15. CMS build lists (continued)

If the build list is:	Then follow this build procedure:
DMSBLPRP	"Rebuild the Load Library" on page 4-15
CMSALOAD CMSCLOAD CMSMLOAD CMSXLOAD DMSBLVBL GSUMLOAD IOCPLOAD SERVLOAD	"Rebuild the Executable MODULEs" on page 4-16
DMSBLFAS	"Rebuild the Assemble Module" on page 4-17
DMSBLVMT DMSBLVML DMSVSLIB GUIBLCSL	"Rebuild the Callable Services Libraries (CSLs)" on page 4-17
DMSSBBAM DMSSBDOS DMSSBGER DMSSBHLP DMSSBINS DMSSBKAN DMSSBPIP DMSSBRTS DMSSBSFS DMSSBSVM DMSSBUCE DMSSBVML GUISBCSL DMSSBAME	"Process the Segment Build Lists" on page 4-18
CMSLOAD	"Rebuild the CMS Nucleus" on page 4-19

Select and Copy Serviced Files

Table 4-16 specifies build lists for CMS that require the select and copy build functions.

Table 4-16. 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
DMSBL990	Build list for sample zCMS for 990 disk	990/990
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

Detailed Build Steps: CMS

Table 4-16. Build lists for CMS select and copy functions (continued)

Build List	Description	Test/Production Disk
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
GSUBLSYS	Build list for Shell non-executable parts	400/400
DMSBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-16 that was identified in the \$VMFBLD \$MSGLOG.

```
vmfbld ppf zvm {cms|cmssfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-16 on page 4-13.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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 4-15.

Table 4-17 contains a list of CMS MACLIBs that may need to be rebuilt.

Table 4-17. 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 zvm {cms|cmssfs} bldlist (serviced

bldlist is the file name of one of the build lists in Table 4-17 on page 4-14.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

vmfview build

Rebuild the Text Libraries

Table 4-18 describes the build lists used to rebuild the CMS TXTLIBs.

Table 4-18. 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
GSUBLLIB	Build files for the LIBC TXTLIB	400/400

1. For each build list in Table 4-18 that has been serviced, enter the following command to create the updated level of the TXTLIB associated with it.

vmfbld ppf zvm {cms|cmssfs} bldlist (serviced

bldlist is the file name of one of the build lists in Table 4-18 on page 4-15.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

vmfview build

Rebuild the Load Library

Table 4-19 contains the load library that may need to be rebuilt.

Table 4-19. 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 zvm {cms|cmssfs} dmsblprp (serviced

Detailed Build Steps: CMS

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

vmfview build

Rebuild the Executable MODULEs

Table 4-20 describes the build lists used to rebuild executable MODULEs.

Table 4-20. Build lists for CMS executable MODULEs

Build List	Description	Test/Production Disk
CMSALOAD	Build list for C-based CMS modules that reside on the system tools disk	493/193
CMSCLOAD	Build list for C-based CMS modules that reside on the system disk	490/190
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
DMSBLVBL	Build list for Binder/Loader MODULEs that reside on the system disk	490/190
GSUMLOAD	Build list for Shell MODULEs	400/400
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 4-20 that has MODULEs that have been serviced, enter the VMFBLD command to rebuild the serviced modules:

vmfbld ppf zvm {cms|cmssfs} bldlist (serviced

bldlist is the file name of one of the build lists in Table 4-20 on page 4-16.

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 message:
DMSLI0201W The following names are undefined: ISPLINK ARIPRDI
disregard it because the names 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:
DMSLI0201W 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:
DMSLI0201W 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 globalled.

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, use online HELP or see the appropriate message book.

vmfview build

Rebuild the Assemble Module

Table 4-21 contains a CMS build list and MODULE that may need to be rebuilt, which requires special processing.

Table 4-21. 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 4-21 to rebuild the F assembler if any of the requisite files have been serviced.

vmfbld ppf zvm {cms|cmssfs} 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, use online HELP or see the appropriate message book.

vmfview build

Rebuild the Callable Services Libraries (CSLs)

Table 4-22 describes the build lists used to rebuild VMMTLIB and VMLIB.

Table 4-22. 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
DMSVSLIB	Build list for DMSVSLIB	493/193
GUIBLCSL	Build list for VMGUILIB	490/190

1. The following command will build the VMLIB CSLLIB, VMLIB CSLSEG, VMGUILIB CSLSEG, VMMTLIB TEXT, and the DMSVSLIB CSLLIB if any of their parts have been serviced.

vmfbld ppf zvm {cms|cmssfs} bldlist (serviced

bldlist is the file name of one of the build lists in Table 4-22.

Note: If you receive the following message while building the CSL libraries disregard it.

DMSSOP2534I No MACLIB libraries have been globalled.

Detailed Build Steps: CMS

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

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 12. Place Your Serviced Components into Production."

Table 4-23. Build lists for CMS segment build lists

Build List	Description
DMSSBBAM	Build list for the CMSBAM saved segment
DMSSBDOS	Build list for the CMSDOS saved segment
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
DMSSBAME	Build list for national language American English

1. Enter the following command for each segment build list, if it has been serviced:

```
vmfbld ppf zvm {cms|cmssfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-23 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, use online HELP or see the appropriate message book.

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.

1. Make sure your machine environment is correct (XA), your storage is defined to at least 256MB 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 zvm {cms|cmssfs} ($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 *z/VM: CMS Planning and Administration* book.

3. Build the nucleus.

```
spool prt *
spool pun *
vmfbld ppf zvm {cms|cmssfs} cmsload (all setup
```

Spool the printer and punch output to your virtual reader.

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.

Detailed Build Steps: CMS

```
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 *z/VM: CMS 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 12. Place Your Serviced Components into Production” on page 3-67.

```
z/VM V5.3.0 mm/dd/yy hh:mm
```

ENTER

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

4. Save the CMS load map.

```
access 493 fm-493
```

```
spool prt nohold
```

```
close prt
```

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

fm-493 is any free file mode.

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 5VMCMS30%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 4-19 and “IPL the CMS Nucleus” on page 4-19. Then return to “Test the New Level of CMS” on page 3-32.

```
xedit cmsnuc map
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found
====> all/undefin/
DMSXDC546E Target not found
====> qquit
```

This response means that no unresolved symbols were found.

This response means that no undefined symbols were found.

6. Redefine the CMS disks back to their original values:

```
define 190 490
define 590 190
ipl 190 clear
z/VM V5.3.0 mm/dd/yy hh:mm
ENTER
```

Press ENTER to complete the IPL.

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 3-32 and complete the rest of the service steps (such as merge processing and placing into production) for this component.

Detailed Build Steps: CP

Prepare to Build

Note

The *compname* is **cp** or **cpsfs**. If you installed CP in the shared file system (SFS), use **cpsfs** as the *compname* instead of **cp**. Moving CP from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**cp|cpsfs**}. Enter **cp** or **cpsfs**. Do not enter the "I".

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 4-24, and then run the corresponding build steps in the order that they appear in the table.

Table 4-24. CP build lists

If the build list is:	Then follow this build procedure:
HCPBL490 HCPBL493 HCPBLHLP HCPBLIMG HCPBLINS HCPBLRAS HCPBLSAM HCPBLSDK HCPBLSRC HCPBLUCE	"Select and Copy Serviced Files" on page 4-23
HCPESAME HCPGPI HCPOM1 HCPOM2 HCPPSI	"Update the Macro Libraries" on page 4-23
HCPMLOAD HCPXLOAD HCQMLOAD	"Rebuild the Executable MODULEs" on page 4-24
HCPSBGER HCPSBHLP HCPSBKAN HCPSBUCE HCPSBAME	"Process the Segment Build Lists" on page 4-24
HCPBLUTL	"Rebuild the CP Utilities" on page 4-25
CPLOAD HCPBLNUC	"Rebuild the CP Nucleus" on page 4-25

Select and Copy Serviced Files

Table 4-25 contains build lists for CP that require the select and copy build functions.

Table 4-25. 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
HCPBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-25 that was in the \$VMFBLD \$MSGLOG.

```
vmfbld ppf zvm {cp|cpsfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-25 on page 4-23.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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 4-24.

Table 4-26 lists the CP MACLIBs that may need to be rebuilt.

Table 4-26. Build lists for CP MACLIBs

Build Lists	Description	Test/Production Disk
HCPESAME	Build list for the HCPESAME MACLIB	493/193
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

Detailed Build Steps: CP

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 zvm {cp|cpsfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-26 on page 4-23.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Executable MODULES

Table 4-27 describes the build lists used to rebuild executable MODULES.

Table 4-27. 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
HCQMLOAD	Build list for VMFBLD to build MODULES for the VM Dump Tool	493/193

1. For each build list in Table 4-27 that has MODULES that have been serviced, enter the VMFBLD command to rebuild the serviced MODULES:

```
vmfbld ppf zvm {cp|cpsfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-27 on page 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, use online HELP or see the appropriate message book.

```
vmfview build
```

Process the Segment Build Lists

This step updates the build status **only**. The actual build of the segment is performed in Step 12. Place Your Serviced Components into Production.

Table 4-28. Build lists for CP segment build lists

Build List	Description
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
HCPSBAME	Build list for national language American English

1. Enter the following command for each segment build list, if it has been serviced.

```
vmfbld ppf zvm {cp|cpsfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-28 on page 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, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the CP Utilities

Table 4-29 shows the build list for CP utilities.

Table 4-29. 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 zvm {cp|cpsfs} hcpblutl (serviced
```

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

```
vmfview build
```

```
BD:VMFBDO2180I HCPBLUTL SA.DUMP RUN HCPSADMP EXEC
```

If this message is shown 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
```

```
z/VM V5.3.0 mm/dd/yy hh:mm
```

```
ENTER
```

HCPSADMP loads a Disabled Wait State PSW when it completes. For more information about this exec, see *z/VM: CP Planning and Administration*. Press ENTER to complete the IPL.

Rebuild the CP Nucleus

Table 4-30 shows the build list for CP nucleus.

Table 4-30. Build list for CP nucleus

Build List	Description	Test/Production Disk
CPLOAD	Build list for CPMOD MODULE	493/193

1. Make sure your machine environment is correct (XA), your storage is defined to at least 256MB 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).

Detailed Build Steps: CP

2. Build the nucleus.

`vmfbld ppf ppfname {cp|cpsfs} cpload (all setup` **cpload** is the CP load list.

Note: If you receive the following:

```
CSECT'S WITH SIZE GREATER THAN CONDITIONAL PAGE BOUNDARY
@MAPSTRT
```

disregard it. For more information, see *z/VM: CP Commands and Utilities Reference*.

3. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`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 maps, see "Rebuild the CP Nucleus" on page 4-25 and Verify the CP Load Map. Then continue with "Merge Processing" on page 3-40.

```
xedit cpload map fm-493
====> set case mixed ignore
====> all/unresol/
DMSXDC546E Target not found
====> all/undefin/
DMSXDC546E Target not found
====> qquit
```

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

This response means that no unresolved symbols were found.

This response means that no undefined symbols were found.

Create a Back-Up Copy of the Primary Parm Disk

Make a copy of the current parm disk.

1. Link and access the CF3 parm disk in write mode.

```
cprelease c
link maint cf3 cf3 m
access cf3 fm3 (erase
```

fm3 is any free filemode

2. Use ACCESSM0 command to allow files with a file mode of '0' to be copied. Then copy all files from the CF1 parm disk to the CF3 parm disk.

```
access 493 r
accessm0 on
access cf1 fm1 (mode0
copyfile * * fm1 = fm3 (olddate replace
```

fm1 is any free filemode.

Create the New CP Nucleus on the Secondary Parm Disk

Make a test parm disk, from the current parm disk and the new CP nucleus, on the Secondary Parm disk.

1. Link and access the CF2 parm disk in write mode.

```
cprelease b
link maint cf2 cf2 m
access cf2 fm2 (erase
```

fm2 is any free filemode.

2. Copy all files from the CF1 parm disk to the CF2 parm disk.

```
copyfile * * fm1 = = fm2 (olddate replace
```

3. Rename the current CPLOAD MODULE and copy the new CPLOAD MODULE.

```
erase cplold module fm2
rename cpload module fm2 cplold = =
copyfile cpload module r = = fm2 (olddate replace
```

Erase the backup CP nucleus.

Copy the new CP nucleus to the parm disk.

4. Reset your environment.

```
release fm3 (detach
cpaccess maint cf3 c rr
release fm2 (detach
cpaccess maint cf2 b rr
release fm1
accessm0 off
release r
```

Note

You have now finished the detailed build steps for CP. Go to “Merge Processing” on page 3-40 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

The *comprname* is **gcs** or **gcssf**s. If you installed GCS in the shared file system (SFS), use **gcssf**s as the *comprname* instead of **gcs**. Moving GCS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**gcs|gcssf**s}. Enter **gcs** or **gcssf**s. 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 4-31, and then run the corresponding build steps in the order that they appear in the table.

Table 4-31. GCS build lists

If the build list is:	Then follow this build procedure:
GCTBLHLP GCTBLINF GCTBLRAS GCTBLRST GCTBLUCE	“Select and Copy Serviced Files” on page 4-28
GCTGPI	“Update the Macro Libraries” on page 4-29
GCTBLPRP	“Rebuild the Load Library” on page 4-29
GCTSBHLP	“Process the Segment Build List” on page 4-30
GCTLOAD	“Rebuild the GCS Nucleus” on page 4-30

Select and Copy Serviced Files

Table 4-32 contains build lists for GCS that require the select and copy build functions.

Table 4-32. 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
GCTBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-32 that was in the \$VMFBLD \$MSGLOG.

```
vmfbld ppf zvm {gcs|gcssf} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-32 on page 4-28.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
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.”

Table 4-33 lists the GCS MACLIBs that may need to be rebuilt.

Table 4-33. Build list for GCS MACLIBs

Build List	Description	Test/Production disk
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 zvm {gcs|gcssf} 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, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Load Library

Table 4-34 contains the load library that may need to be rebuilt.

Table 4-34. 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 zvm {gcs|gcssf} gctblprp (serviced
```

Detailed Build Steps: GCS

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

vmfview build

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed in Step 12. Place Your Serviced Components into Production.

Table 4-35. 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 zvm {gcs|gcssf} 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, use online HELP or see the appropriate message book.

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.

1. Make sure your machine environment is correct (XA), your storage is defined to at least 256MB 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 *z/VM: Group Control System*.
3. Build the nucleus.

```
spool prt *
spool pun *
vmfbld ppf ppfname {gcs|gcssf} gctload (all setup
```

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

```
SYSTEM LOAD DECK COMPLETE
PUN FILE spoolid TO MAINT COPY 001 NOHOLD
```

gctload is the build list for the GCS nucleus. If you are not using the default, substitute your nucleus name for **gctload**.

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
```

```
NAMED SAVED SYSTEM (NSS) GCS
WAS SUCCESSFULLY SAVED IN FILEID 0100
RDR FILE spoolid SENT FROM MAINT
```

Save the nucleus in your saved segment.

This message indicates that your nucleus was successfully saved.

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

Restore your CMS environment.

```
ipl 190 clear
```

```
z/VM V5.3.0 mm/dd/yy hh:mm
```

```
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 5VMGCS30%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 4-30 and “IPL the GCS Nucleus.” Then return to “Merge Processing” on page 3-48.

```
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.

Detailed Build Steps: GCS

Note

You have now finished the detailed build steps for GCS. Go to “Merge Processing” on page 3-48 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

Prepare to Build

Note

The *compname* is **dv** or **dvsfs**. If you installed the Dump Viewing Facility in the shared file system (SFS), use **dvsfs** as the *compname* instead of **dv**. Moving the Dump Viewing Facility from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component names is {**dv|dvsfs**}. Enter **dv** or **dvsfs**. 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. Dump Viewing Facility build lists

If the build list is:	Then follow this build procedure:
HCSBL493 HCSBLHLP HCSBLMSG HCSBLSRC HCSBLUCE	"Select and Copy Serviced Files" on page 4-33
HCSMLOAD HCSSLLOAD	"Rebuild the Executable MODULEs" on page 4-34
HCSSBHLP	"Process the Segment Build List" on page 4-34

Select and Copy Serviced Files

Table 4-37 contains build lists for Dump Viewing Facility that require the select and copy build functions.

Table 4-37. 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
HCSBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-37 that was in the \$VMFBLD \$MSGLOG.

Detailed Build Steps: Dump Viewing Facility

`vmfbld ppf zvm {dv|dvsfs} bldlist (serviced`

bldlist is the file name of one of the build lists in Table 4-37 on page 4-33.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`vmfview build`

Rebuild the Executable MODULES

If any Dump Viewing Facility executable MODULES have been serviced, they must be rebuilt.

Table 4-38. 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 zvm {dv|dvsfs} bldlist (serviced`

bldlist is the file name of one of the build lists in Table 4-38 on page 4-34.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`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 12. Place Your Serviced Components into Production.

Table 4-39. 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 zvm {dv|dvsfs} hcsshlp (serviced`

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

`vmfview build`

Note

You have now finished the detailed build steps for Dump Viewing Facility. Go to “Merge Processing” 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

The *compname* is **tsaf** or **tsafsfs**. 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 z/VM installation process.

In the following procedures, 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 4-40, and then run the corresponding build steps in the order that they appear in the table.

Table 4-40. TSAF build lists

If the build list is:	Then follow this build procedure:
ATSBHLHP ATSBLINF ATSBLMES ATSBLSRC ATSBLUCE	“Select and Copy Serviced Files” on page 4-36
ATSMLOAD	“Rebuild the Executable MODULEs” on page 4-37
ATSSBHLP	“Process the Segment Build List” on page 4-37

Select and Copy Serviced Files

Table 4-41 contains build lists for TSAF that require the select and copy build functions.

Table 4-41. Build lists for TSAF select and copy functions

Build List	Description	Test/Production Disk
ATSBHLHP	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
ATSBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-41 that was in the \$VMFBLD \$MSGLOG.

`vmfbld ppf zvm {tsaf|tsafsfs} bldlist (serviced` *bldlist* is the file name of one of the build lists in Table 4-41 on page 4-36.

- 2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. Form information about handling specific build messages, use online HELP or see the appropriate message book.

`vmfview build`

Rebuild the Executable MODULEs

Table 4-42 describes the build list used to rebuild executable MODULEs.

Table 4-42. 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 zvm {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, use online HELP or see the appropriate message book.

`vmfview build`

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed in Step 12. Place Your Serviced Components into Production.

Table 4-43. 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 zvm {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, use online HELP or see the appropriate message book.

`vmfview build`

Note

You have now finished the detailed build steps for TSAF. Go to “Merge Processing” on page 3-59 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

The *comname* is **avs** or **avssfs**. If you installed AVS in the shared file system, use **avssfs** as the *comname* instead of **avs**. Moving AVS from a CMS minidisk to an SFS directory was decided during the z/VM installation process.

In the following procedures, the notation for the component name is {**avslavssfs**}. 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 4-44, and then run the corresponding build steps in the order that they appear in the table.

Table 4-44. AVS build lists

If the build list is:	Then follow this build procedure:
AGWBLHLP AGWBLINF AGWBLSRC AGWBLUCE	“Select and Copy Serviced Files” on page 4-38
AGWBLLIB AGWBLUTL	“Rebuild the Load Libraries” on page 4-39
AGWSBHLP	“Process the Segment Build List” on page 4-39

Select and Copy Serviced Files

Table 4-45 contains build lists for AVS that require the select and copy build functions.

Table 4-45. 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
AGWBLUCE	Build list for Uppercase HELP files	402/402

1. Enter the following command for each build list in Table 4-45 that was in the \$VMFBLD \$MSGLOG.

```
vmfbld ppf zvm {avs|avssfs} bldlist (serviced
```

bldlist is the file name of one of the build lists in Table 4-45 on page 4-38.

2. Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, use online HELP or see the appropriate message book.

```
vmfview build
```

Rebuild the Load Libraries

Table 4-46 lists the AVS load libraries that may need to be rebuilt.

Table 4-46. 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 zvm {avs|avssfs} agwblutl (serviced
```

2. To update the AGW LOADLIB, enter the following VMFBLD command.

```
vmfbld ppf zvm {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, use online HELP or see the appropriate message book.

```
vmfview build
```

Process the Segment Build List

This step updates the build status **only**. The actual build of the segment is performed in Step 12. Place Your Serviced Components into Production.

Table 4-47. 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 zvm {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, use online HELP or see the appropriate message book.

```
vmfview build
```

Detailed Build Steps: AVS

Note

You have now finished the detailed build steps for AVS. Go to “Merge Processing” on page 3-65 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 z/VM components. Local service and local modifications are defined as any service that is applied to your z/VM 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 can create 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 z/VM 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 a customer makes to tailor their z/VM system. These updates can be supplied by IBM licensed products or by other vendor products. For example, if you have RACF® Security Server for z/VM, you have a customer local modification because RACF 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. Finally, 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.

A Rework of 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 a 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:

<i>aparnumber</i>	<i>aparnumber</i> is the APAR number.
<i>appid</i>	<i>appid</i> 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 5-5.
<i>auxft</i>	<i>auxft</i> 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 5-5.
<i>cftabbrev</i>	<i>cftabbrev</i> is the abbreviation for the compiled file type: CEX for EXEC files or CAE for XEDIT files.
<i>comment</i>	<i>comment</i> is any comment that you want to add when you XEDIT a file.

Local Service and Modifications — Overview

<i>compname</i>	<i>compname</i> is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, LE, CMS, CP, GCS, DV, TSAF, and AVS.
<i>ctlfile</i>	<i>ctlfile</i> 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 5-5.
<i>fm-local</i>	<i>fm-local</i> is the file mode of the LOCALMOD disk for the component you are servicing.
<i>fn</i>	<i>fn</i> is the file name of the part.
<i>ft</i>	<i>ft</i> is the file type of the part.
<i>ftabbrev</i>	<i>ftabbrev</i> is the 3 character abbreviation for the file type of the part that was modified. To obtain a <i>ftabbrev</i> see “Obtaining File Type Abbreviations” on page 5-5.
<i>modid</i>	<i>modid</i> 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 <i>modid</i> is also used to create the file type of the object generated when the local modification is compiled.
<i>curmodid</i>	<i>curmodid</i> is the current local modification number.
<i>ftabbrev-modid</i>	<i>ftabbrev-modid</i> is the file type of the compiled part and consists of the <i>ftabbrev</i> concatenated with the <i>modid</i> . The <i>ftabbrev</i> is the 3 character file type abbreviation. See the <i>modid</i> abbreviation for more information.
<i>ftabbrev-ptfnumber</i>	<i>ftabbrev-ptfnumber</i> is the file type of the IBM replacement service to a part and consists of the <i>ftabbrev</i> concatenated with the <i>ptfnumber</i> . The <i>ftabbrev</i> is the 3 character file type abbreviation. The <i>ptfnumber</i> is the PTF number.
<i>lcmoid</i>	<i>lcmoid</i> is lc concatenated with <i>modid</i> and is the local modification identification. The first 2 characters should be LC, followed by the <i>modid</i> . For example, LCLFIX1 is valid. This must be in uppercase.
<i>ppfname</i>	<i>ppfname</i> is the file name of the PPF being used. <i>ppfname</i> is shown as zvm . If you have your own <i>ppfname</i> , substitute that name for the zvm name shown in the commands.
<i>ptfnumber</i>	<i>ptfnumber</i> is the PTF number.
<i>realft</i>	<i>realft</i> is the real file type of the part. For example, MACRO or COPY.
<i>sftabbrev</i>	<i>sftabbrev</i> is the abbreviation for the uncompiled file type of a compiled part: SXE for EXEC files or SXD for XEDIT files.
<i>svclvl</i>	<i>svclvl</i> is a service level indicator. This must be in uppercase. This is the second token in the AUX file entries and is usually LCL.
<i>updateft</i>	<i>updateft</i> is the file type of the update file containing your local modifications to source files. This must be in uppercase. <i>updateft</i> is not used for replacement parts. You will need to define this if the update file was not supplied by IBM software support.
<i>vvft</i>	<i>vvft</i> 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 5-5.

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	5VMSES30	AUXLCL	VVTLCL
REXX/VM	IXXVM	5VMREX30	AUXLIXX	VVTLIXX
Language Environment	CEEVM	4VMVMQ40	AUXLCL	VVTLCL
CMS	DMSVM	5VMCMS30	AUXLCL	VVTLCL
CP	HCPVM	5VMCPR30	AUXLCL	VVTLCL
GCS	GCTVM	5VMGCS30	AUXLGCT	VVTLGCT
Dump Viewing Facility	HCSVM	5VMDVF30	AUXLCL	VVTLCL
TSAF	ATSVM	5VMTSA30	AUXLCL	VVTLCL
AVS	AGWVM	5VMAVS30	AUXLCL	VVTLCL

Note: When assembling CMS Pipelines parts (any part with a file name of FPLxxx 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 *z/VM: Guide for Automated Installation and Service* for instructions on storing a backup copy of the z/VM system to tape.
2. Set up the correct virtual machine environment and IPL.

```
set machine xa
System reset.
System = XA
define storage nnM
STORAGE CLEARED - SYSTEM RESET
STORAGE =      nnM
ipl 190 clear
```

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

Define the storage to at least 256MB.

Defining storage causes a system reset.

z/VM V5.3.0 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 zvm compname
```

zvm is the PPF file name. If you have an override to **zvm**, use it instead.

compname is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, LE, 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 zvm compname
```

Procedures for Local Service and Modifications

6. 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*.
7. 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 5-5 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 6-6 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.
8. 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. For LE, the command must be issued twice, once for each of the two *bldlists* listed.

vmfbld ppf zvm 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
LE	EDCBLFPR
LE	CEEBLFPR

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 6-6.

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 6-7.

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.

Procedures for Local Service and Modifications

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 lmodid comment
====> file
```

Step 3b. Create or Copy the Update File

Note

You must complete either substep 1 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 6-7.

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” on page 6-5.

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 chk1vl zvm 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 zvm 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 6-10

3. Update the \$SELECT file for the macro.

```
vmfrep1 fn ft zvm 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, 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 lmodid 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 ./ * lmodid 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 6-7.

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.

```
vmfrep1 fn ft zvm 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 ZVM 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 6-7.

2. If you have the replacement part, copy the file to the local modification disk.

```
vmfrep1 fn ft zvm 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 ZVM 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 zvm 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 6-10.

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 lmodid 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 ./ * lmodid 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 6-6. Then continue with “Step 5. Rebuild Objects.”

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 6-6.

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.) For LE, the command must be issued twice, once for each of the two *bldlists* listed.

vmfbld ppf zvm 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
LE	EDCBLFPR
LE	CEEBLFPR

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 zvm 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.

```
vmfrep1 fn ft zvm compname ($select ftabbr cftabbrev logmod modid nocopy outmode localmod
```

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.

```
vmfrep1 fn ft zvm 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 zvm 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 ftabbr-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.

Procedures for Local Service and Modifications

- 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 zvm cp (nohist outmode localmod $select logmod
```

If for some reason the VMFEXUPD command is unsuccessful, return to “Step 3b. Create or Copy the Update File” on page 6-4 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 6-6 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 zvm compname bldlist (serviced
```

bldlist is the file name of one of the build lists 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 6-6.

For local modifications to the REXX/VM component, complete substep 2; **for all other components**, complete substep 1.

1. For components besides REXX/VM use the following command for ASSEMBLE files, except for the DMSTRTcc ASSEMBLE file. For the DMSTRTcc ASSEMBLE file, (where cc is a country code suffix), see “Step 5g. Create a Replacement Part from Modified National Language Files” on page 6-12. 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 FPLxxx 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 FPLVM option.

```
vmfhlasm fn zvm compname (logmod $select outmode localmod
```

OR

```
vmfhlasz fn zvm cms (logmod $select outmode localmod cntrl fp1vm
```

Continue with “Step 5f. Create a Replacement Part from Replaced ASSEMBLE Files.”

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 chk1vl zvm 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.

```
vmfhlasz fn zvm cms ($select outmode localmod
```

- c. In “Step 5h. Rebuild Remaining Objects” on page 6-12 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 6-6.

1. Enter the following command to update the local version vector table for the assembled text file.

```
vmfrep1 fn text zvm 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 6-6.

2. Enter the VMFHLASM command to assemble the file. The command places the output file on the LOCALMOD disk and updates the \$SELECT file.

```
vmfhlasz fn zvm compname (nckgen $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).

For local modifications to the CP component, complete substeps 1 and 2; for all other components, complete subset 1 only.

For all components, 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 zvm 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 *z/VM: 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 3-9
REXX/VM	"Prepare to Build" on page 3-16
Language Environment	"Prepare to Build" on page 3-21
CMS	"Prepare to Build" on page 3-28
CP	"Prepare to Build" on page 3-37
GCS	"Prepare to Build" on page 3-45
Dump Viewing Facility	"Prepare to Build" on page 3-52
TSAF	"Prepare to Build" on page 3-57
AVS	"Prepare to Build" on page 3-64

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 *z/VM: Guide for Automated Installation and Service* for instructions on storing a backup copy of the z/VM system to tape.
2. Set up the correct virtual machine environment and IPL.

```
set machine xa
System reset.
System = XA
define storage nnM
STORAGE CLEARED - SYSTEM RESET
STORAGE =      nnM
ipl 190 clear
```

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

Define the storage to at least 256MB.

Defining storage causes a system reset.

z/VM V5.3.0 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 zvm compname
```

zvm is the \$PPF. If you have an override to **zvm**, use it instead.

compname is the default component identifier, unless you use your own override file. The default component identifiers are VMSES, REXX, LE, 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 zvm compname
```

Procedures for Reworking Local Service and Modifications

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. For LE, the command must be issued twice, once for each of the two *bldlists* listed.

vmfbld ppf zvm 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
LE	EDCBLFPR
LE	CEEBLFPR

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.

Procedures for Reworking Local Service and Modifications

vmfqobj **zvm compname tdata :object fn :libname**

fn is the name of the MACRO that needs to be reworked.

:OBJECT *bldlist.fn*
:LIBNAME *libname*

Use VMFQOBJ to determine what MACLIBs need to be rebuilt.

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.

vmfrep1 *fn ft zvm 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.

vmfrep1 *fn ft zvm 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.

vmfrep1 *fn ft zvm compname (\$select nocopy*

4. If you are reworking a MACRO, use VMFQOBJ to determine what MACLIBs need to be rebuilt.

vmfqobj **zvm 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 7-6.

Step 4. Rebuild Objects

Note

When you have completed the preceding steps for all the rework to the local modifications for this component 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.) For LE, the command must be issued twice, once for each of the two *bldlists* listed.

```
vmfbld ppf zvm 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
LE	EDCBLFPR
LE	CEEBLFPR

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 zvm compname (outmode localmod $select ftabbr sftabbrev
```

Procedures for Reworking Local Service and Modifications

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. Update the \$SELECT file. Skip substep 3.

```
vmfrep1 fn ft zvm 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.

```
vmfrep1 fn ft zvm 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 zvm 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 ftabbre-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.**

Procedures for Reworking Local Service and Modifications

- 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 zvm 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 zvm 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 DMSTRTcc ASSEMBLE file. For the DMSTRTcc ASSEMBLE file, (where *cc* is a country code suffix), see “Step 4g. Create a

Procedures for Reworking Local Service and Modifications

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 FPLxxx 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 FPLVM option.

```
vmfhlas fn zvm compname ($select outmode localmod
```

OR

```
vmfhlas fn zvm 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.

```
vmfhlas fn zvm cms ($select outmode localmod
```

- b. In “Step 4h. Rebuild Remaining Objects” on page 7-8 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.

```
vmfhlas fn zvm 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).

For local modifications to the CP component, complete substeps 1 and 2; for all other components, complete substep 1 only.

For all components, 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 zvm 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 *z/VM: 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 3-9
REXX/VM	“Prepare to Build” on page 3-16
Language Environment	“Prepare to Build” on page 3-21
CMS	“Prepare to Build” on page 3-28
CP	“Prepare to Build” on page 3-37
GCS	“Prepare to Build” on page 3-45
Dump Viewing Facility	“Prepare to Build” on page 3-52
TSAF	“Prepare to Build” on page 3-57
AVS	“Prepare to Build” on page 3-64

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. VMFREM can also remove local modifications.

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, single PTFs, and local modifications.

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 zvm compname`

2. Un-apply the current service level or un-apply the current and the test service levels.

`vmfrem ppf zvm 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)

Removing Service

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 6-12 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 zvm compname`

2. Un-apply the PTF and its dependents.

`vmfrem ppf zvm 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 6-12 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 zvm compname
```

3. Follow the build steps identified in Table 6-1 on page 6-12 for the component being processed. After all build steps have been completed for the component, all service will have been restored.

Removing A Local Modification

If you want to back out a local modification, you must perform the following steps. Any local modifications that are dependent on the specified local modification will also be backed out.

1. Access the service disks required for the component using the VMFSETUP command.

```
vmfsetup zvm compname
```

2. Remove the local modification.

```
vmfrem ppf zvm compname mod modid
```

modid is the one- to five-character local modification number of the local modification you are removing.

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 MODs.
```

If you respond to continue, you must restore the missing parts before running VMFBLD. You can restore the missing parts 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.
- Recreate them, if they are local modification parts.

3. Follow the build steps identified in Table 6-1 on page 6-12 for the component being processed. After all build steps have been completed for the component, the specified local modification and its dependents will have been removed.

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.

Determine the Receive Steps to Follow

1. Follow the procedures that you received electronically, in a README file or in the *ServiceLink User's Guide*, to retrieve and decompress the envelope file on your A-disk. You will run DETERSE to decompress the envelope. The file type of the decompressed envelope file must be SERVLINK.
2. Determine whether the service files are in VMSES or non-VMSES format.
 - If the file name and extension of the electronically delivered service documentation envelope or PTF envelope file is one of the following, it is in VMSES format.

If the file name and extension are...	Go to
VLSTn...n.bin ordernum.SHIPDOCS	"Receive Service Documentation"
VPTFn...n.bin RPTFn...n.bin ordernum.SHIPTFSS ordernum.SHIPRSUn	"Receive the Service" on page A-2
RLSTn...n.bin ordernum.SHIPDOC	No further steps are necessary, as these are readable text files.

- If the file name and extension of the electronically delivered service documentation envelope or PTF envelope file is one of the following, it is in non-VMSES format. Use the steps in Appendix B, "Receiving Service For Non-VMSES Envelopes," on page B-1 to receive the documentation and service instead.

If the file name and extension are...	Go to
NLSTn...n.bin ordernum.SHIPDOCN	Step 1 on page B-1
NPTFn...n.bin ordernum.SHIPTFSN	Step 2 on page B-1

Receive Service Documentation

1. Receive the service documentation from the documentation envelope *docenvfn* 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.

Receiving Service for VMSES Envelopes

- b. To create a SERVICE DISKMAP file for the corresponding PTF envelope, enter the following command and substitute *ptfenvfn* for *docenvfn*.

vmfrec info (*env docenvfn*

where *docenvfn* is the file name of the service documentation envelope (SERVLINK) file.

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 z/VM component(s) that are in the envelope.
2. 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, use online HELP or see the appropriate message book.

vmfview receive

3. 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 z/VM from the PTF envelope *ptfenvfn* SERVLINK.

vmfrec ppf zvm compname (*env ptfenvfn* **setup**

zvm 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 **zvm**. The file type must be PPF.

compname is the name of the component. *ptfenvfn* is the file name of the service PTF envelope (SERVLINK) file.

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, use online HELP or see the appropriate message book.

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 *nsdocenvfn* SERVLINK.

```
vmfplcd rst env= nsdocenvfn 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 *nsptfenvfn* SERVLINK.

```
vmfplcd rst env= nsptfenvfn 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.

Receiving Service For Non-VMSES Envelopes

Appendix C. Building the NLS Nuclei, Including UCENG

The preferred way to change the default system language for CP, CMS, and GCS is to set it in the system configuration file by using the `USER_DEFAULTS` statement or by selecting a system language when you install z/VM. See *z/VM: CP Planning and Administration*. Previously this was done by rebuilding the nuclei to contain the NLS files.

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. VMSESRVS 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 (VMSESRVS, VMSESRVR, and VMSESRVU) 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).

z/VM V5.3.0 mm/dd/yy hh:mm

**** DO NOT press ENTER! ****

access (nosprof

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 VMSESRVS 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.

Test the New Level of CMS

```
link maint 490 190 rr
link maint 493 193 rr
ipl 190 clear parm nosprof instseg no
```

Link the test CMS system disk as 190.
Link the test system tools disk as 193.
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).

z/VM V5.3.0 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 193
define 590 190
define 593 193
ipl 190 clear
z/VM V5.3.0 mm/dd/yy hh:mm
disconnect
```

Detach the test CMS system disk.
Detach the test system tools disk.
Define the production CMS system disk as 190.
Define the production system tools disk as 193.
IPL the old (production) level of CMS.
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

stop
detach 190
detach 193
define 590 190
define 593 193
ipl 190 clear
z/VM V5.3.0 mm/dd/yy hh:mm
```

Log on the SFS server user ID used to test the new level of CMS.
Stop the server.
Detach the test CMS system disk.
Detach the test system tools disk.
Define the production CMS system disk as 190.
Define the production system tools disk as 193.
IPL the old (production) level of CMS.

**** DO NOT press ENTER! ****

```
access (nosprof
access 193 b
xedit userid dmssparms
====> top
====> set case mixed ignore
====> locate/nosavesegid
====> change/NOSAVESEGID/SAVESEGID savesegname

====> file
fileserv start
#cp disconnect
```

Bypass the execution of the PROFILE EXEC.
Access the production system tools disk.
XEDIT the DMSPARMS file for the server.
Start at the top of the file.
Insure that the search is not case significant.
Locate the NOSAVESEGID parameter.
Change the NOSAVESEGID parameter to SAVESEGID *savesegname*. If you are using an IBM-supplied server (VMSERVS, VMSERVER or VMSERVU), *savesegname* is CMSFILES.
Save the updated DMSPARMS file.
Start the server.
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.

Test the New Level of CMS

Appendix E. Local Modification Example for CMSINST

CMS improves system performance and storage usage by placing heavily used EXECs and MODULES 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 C89 EXEC is added to CMSINST. C89 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 *z/VM: 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 segblst

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 zvm {cms|cmssfs}
```

b. Select a local modification id for this change.

3. Apply local service to CMSINST LSEG.

a. Copy the highest level of CMSINST LSEG to the LOCALMOD disk.

seg10012 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.

```
xedit cmsinst segl0012 e
====> input EXEC C89 EXEC      *
====> file
```

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 C89 EXEC. This will ensure that when C89 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 aux1cl 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 C89 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.C89.EXEC

Add the build requisite for C89 EXEC. DMSBLCRX is the build list that contains C89 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. Because 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 5VMCMS30 VVTLC.

```
vmfexupd dmssbins exec zvm {cms|cmssfs} ($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 zvm {cms|cmssfs} dmssbins (serviced
```

- b. Rebuild the INSTSEG saved segment which contains the CMSINST logical saved segment. See "Rebuild the Saved Segments" on page 3-67.

Local Modification Example for CMSINST

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 *z/VM: CP Exit Customization* for how to update a xxxMDLAT MACRO.

The CP load list for z/VM, 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.

All of the modules that make up the CP nucleus are resident (nonpageable). They also have one or more of these attributes:

- Multiprocessor-capable (the default) or non-multiprocessor-capable. Multiprocessor-capable modules can execute simultaneously on two or more processors. Non-multiprocessor-capable modules can execute only on a single processor, the master processor, which is usually the IPL processor.
- DATA. These modules are data or work area modules.
- Short register (the default) or long register. Short-register modules use 32-bit registers. Long-register modules use 64-bit registers.
- 31-bit addressing mode (the default) or 64-bit addressing mode.
- Primary Space mode (the default) or Access Register mode.

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 module entry points are:

- Dynamic savearea or static
- Traced call and return for dynamic linkage or not traced
- Nonrestricted (can be called directly using HCPCALL) or restricted (can be called only by using a special-purpose cover macro)
- Short register or long register
- 31-bit addressing mode or 64-bit addressing mode
- Primary Space mode or Access Register mode

Some entry point attributes have module defaults. Other entry point attributes have implied module defaults based on the specified or default module attributes.

For more information about CP module attributes and entry point attributes, see the description of the MDLATENT MACRO in *z/VM: CP Exit Customization*.

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 need to update the CP load list, then use these instructions. This involves:

Updating the CP Load List

- 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
- Rebuilding the CP nucleus

Steps to Follow

1. If you need to, review the recommended guidelines for using update files on page 6-3.
2. Modify HCPMDLAT MACRO using a local modification. Because 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:

<i>auxft</i>	AUXLCL
<i>compname</i>	CPICPSFS
<i>fm-local</i>	E
<i>fn</i>	HCPMDLAT
<i>modid</i>	L0001
<i>ppfname</i>	ZVM
<i>svclvl</i>	LCL
<i>updateft</i>	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 zvm {cp|cpsfs}
```

- 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. Either make modifications to the HCPMDLAT MACRO or if you only need to pick up a xxxMDLAT MACRO, then create a dummy update to the HCPMDLAT MACRO.

- 1) To make modifications to the HCPMDLAT MACRO:
 - 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.

- File your changes.

```
====> file = = e
```

- 2) To create a dummy update for the HCPMDLAT MACRO:

```
xedit hcpmdlat m10001hp e (noprof
====> input ./ * 1c10001 dummy update
====> file
```

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 5VMCPR30 \$SELECT file will be updated with two records (HCPMDLAT MACRO and CPLOAD EXEC). The modification to HCPMDLAT MACRO and CPLOAD EXEC will be identified in 5VMCPR30 VVTLC.

```
gencpb1s cpload zvm {cp|cpsfs} (outmode localmod $select logmod mdlat all
```

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 zvm {cp|cpsfs} 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 4-25 and “Verify the CP Load Map” on page 4-26.

Updating the CP Load List

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 4-19 and “IPL the CMS Nucleus” on page 4-19.

CPLOAD then rebuild and save the CP load map. See “Rebuild the CP Nucleus” on page 4-25 and “Verify the CP Load Map” on page 4-26.

GCTLOAD then rebuild and IPL the GCS nucleus. See “Rebuild the GCS Nucleus” on page 4-30 and “IPL the GCS Nucleus” on page 4-31.

- 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 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*

To rebuild the segment, see the appropriate VMFBLD command for the affected segment in “Rebuild the Saved Segments” on page 3-67.

Using VMFBLD Outside of the Service Process

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.

vmfrep1 *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 *z/VM: VMSES/E Introduction and Reference*.

Enter the ZAPTEXT command. For information about the format of this command, see *z/VM: CMS Commands and Utilities Reference*.

For MODULEs, TXTLIBs and LOADLIBs

Create the ZAP input control records to define your changes. For more information about creating these records, see *z/VM: VMSES/E Introduction and Reference*.

Enter the ZAP command. For information about the format of this command, see *z/VM: VMSES/E Introduction and Reference*.

Apply Changes Directly to Object Code

3. Use VMFREPL to copy the modified object to the local disk, update the local version vector table, and update the \$SELECT file.

```
vmfrep1 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 ZVM 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 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

Applying and Removing Patches

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. z/VM Build Lists

The following table identifies the build lists used in servicing z/VM and describes their functions. You can view the contents of each build list using XEDIT.

Table J-1. z/VM 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.
VMFBLUCE	VMSES/E	402/402	This file is input to VMFBLD to copy VMSES/E replacement Uppercase HELP files to the HELP disk.
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.
CEEBLFPR	Language Environment	49E/19E	This file is input to VMFBLD to copy Language Environment "CEL" files to the system product disk.
CEEBLMKD	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEELKED TXTLIB.
CEEBLMAC	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEEMAC MACLIB.
CEEBLMOD	Language Environment	49E/19E	This file is input to VMFBLD to build "CEL" modules on the system product disk.
CEEBLNLS	Language Environment	49E/19E	This file is input to VMFBLD to build NLS modules on the system product disk.
CEEBLOBJ	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEEOBJ TXTLIB.
CEEBLRUN	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEERUN LOADLIB.
CEEBSAM	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEESAMP MACLIB.

z/VM Build Lists

Table J-1. z/VM Build Lists (continued)

Build List	Component	Target	Description/Function
CEEBSLB2	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEEBND2 TXTLIB.
CEEBSLGA	Language Environment	n/a	This file is input to VMFBLD to update the build status for the SCEEX segment.
CEEBSLGB	Language Environment	n/a	This file is input to VMFBLD to update the build status for the SCEE segment.
EDCBL49B	Language Environment	49B/49B	This file is input to VMFBLD to copy locale information to the customization disk.
EDCBLCPP	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEECPP TXTLIB.
EDCBLFPR	Language Environment	49E/19E	This file is input to VMFBLD to copy Language Environment C files to the system product disk.
EDCBLH	Language Environment	49E/19E	This file is input to VMFBLD to copy Language Environment C/VM header files to the system product disk.
EDCBLHPE	Language Environment	19D	This file is input to VMFBLD to copy Language Environment help files to the help disk.
EDCBLHPU	Language Environment	402	This file is input to VMFBLD to copy Language Environment uppercase help files to the help disk.
EDCBLKLD	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEELKD TXTLIB.
EDCBLMAC	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEEMAC MACLIB.
EDCBLMOD	Language Environment	49E/19E	This file is input to VMFBLD to build C modules on the system product disk.
EDCBLPMB	Language Environment	49E/19E	This file is input to VMFBLD to build C modules with the binder/loader on the system product disk.
EDCBLRUN	Language Environment	49E/19E	This file is input to VMFBLD to build the SCEERUN LOADLIB.
EDCBLSB2	Language Environment	49E/19E	This file is input to VMFBLD to build SCEEBND2 TXTLIB.
EDCBLSP1	Language Environment	49E/19E	This file is input to VMFBLD to build SCEESPC TXTLIB.
EDCBLSP2	Language Environment	49E/19E	This file is input to VMFBLD to build SCEESPCO TXTLIB.
EDCSBHLP	Language Environment	n/a	This file is input to VMFBLD to update the build status for the HELP segment.
IBMBLCAL	Language Environment	49E/49E	This file is input to VMFBLD to build the SIBM CALL TXTLIB.
IBMBLFPR	Language Environment	49E/49E	This file is input to VMFBLD to copy COBOL files to the system product disk.
IBMBLLKD	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEELKD TXTLIB.
IBMBLMOD	Language Environment	49E/49E	This file is input to VMFBLD to build the PL/I modules.
IBMBLMTH	Language Environment	49E/49E	This file is input to VMFBLD to build the SIBMMATH TXTLIB.

Table J-1. z/VM Build Lists (continued)

Build List	Component	Target	Description/Function
IBMBLNLS	Language Environment	49E/49E	This file is input to VMFBLD to build the NLS modules.
IBMBLRUN	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEERUN LOADLIB.
IBMBLSAM	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEESAMP MACLIB.
IGZBLFPR	Language Environment	49E/49E	This file is input to VMFBLD to copy PL/I files to the system product disk.
IGZBLLBO	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEELBO LOADLIB.
IGZBLKLD	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEELKED TXTLIB.
IGZBLMAC	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEEMAC MACLIB.
IGZBLMOD	Language Environment	49E/49E	This file is input to VMFBLD to build the COBOL modules.
IGZBLNLS	Language Environment	49E/49E	This file is input to VMFBLD to build the NLS modules.
IGZBLPAC	Language Environment	49E/49E	This file is input to VMFBLD to build the COBOL modules.
IGZBLRUN	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEERUN LOADLIB.
IGZBLSAM	Language Environment	49E/49E	This file is input to VMFBLD to build the SCEESAMP 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.
CMSCLOAD	CMS	490/190	This file is input to VMFBLD to build the C-based CMS modules on the system disk.
CMSALOAD	CMS	493/193	This file is input to VMFBLD to build the C-based CMS modules on the system tools disk.
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.
DMSBL990	CMS	990	This file is input to VMFBLD to copy zCMS replacement files to the zCMS 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.

z/VM Build Lists

Table J-1. z/VM Build Lists (continued)

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.
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.
DMSBLUCE	CMS	402/402	This file is input to VMFBLD to copy CMS replacement Uppercase HELP files to the HELP disk.
DMSBLVBL	CMS	490/190	This file is input to VMFBLD to build Binder/Loader modules on the system 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.
DMSSBAME	CMS	402/402	This file is input to VMFBLD to update the build status for the American English segment.
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.
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.
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.

Table J-1. z/VM Build Lists (continued)

Build List	Component	Target	Description/Function
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.
DMSVSLIB	CMS	493/193	This file is input to VMFBLD to build the DMSVSLIB CSLLIB.
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.
GUISBCSL	CMS	N/A	This file is input to VMFBLD to update the build status for the GUICSLIB segment.
GSUBLLIB	CMS	400/400	This file is input to VMFBLD to build the LIBC TXTLIB.
GSUBLSYS	CMS	400/400	This file is input to VMFBLD to build Shell non-executable parts.
GSUMLOAD	CMS	400/400	This file is input to VMFBLD to build Shell CMS modules.
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 CPLOAD MODULE.
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.
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.

z/VM Build Lists

Table J-1. z/VM Build Lists (continued)

Build List	Component	Target	Description/Function
HCPBLSRC	CP	493/193	This file is input to VMFBLD to copy CP replacement source files to the system tools disk.
HCPBLUCE	CP	402/402	This file is input to VMFBLD to copy CP replacement Uppercase HELP files to the HELP disk.
HCPBLUTL	CP	490/190	This file is input to VMFBLD to build CP utilities on the system disk.
HCPESAME	CP	493/193	This file is input to VMFBLD to build the HCPESAME MACLIB.
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.
H CPRP	CP	493/193	This file is input to VMFBLD to build the H CPRP MACLIB.
HCPXLOAD	CP	493/193	This file is input to VMFBLD to build CP modules on the system tools disk.
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.
HCPSBAME	CP	402/402	This file is input to VMFBLD to update the build status for the American English segment.
HCPSBUCE	CP	N/A	This file is input to VMFBLD to update the build status for the Uppercase English segment.
HCQMLOAD	CP	493/193	This file is input to VMFBLD to build modules for the VM Dump Tool.
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.
GCTBLUCE	GCS	402/402	This file is input to VMFBLD to copy GCS replacement Uppercase HELP files to the HELP 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. z/VM Build Lists (continued)

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.
HCSBLUCE	Dump Viewing Facility	402/402	This file is input to VMFBLD to copy Dump Viewing Facility replacement Uppercase HELP files to the HELP 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.
ATSBLUCE	TSAF	402/402	This file is input to VMFBLD to copy TSAF replacement Uppercase HELP files to the HELP 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.
AGWBLUCE	AVS	402/402	This file is input to VMFBLD to copy AVS replacement Uppercase HELP files to the HELP 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 the Component Service Level

In addition, CMS, CP and GCS also provide commands to query the RSU service level for that component.

Determine the CMS Service Level

In the example, 000 represents the RSU service level (0000).

```
query cmslevel  
CMS Level 23, service level 000
```

Determine the CP Service Level

In the example, 0000 is the RSU service level.

```
query cplevel  
z/VM Version 5 Release 3.0, service level 0000 (64-bit)  
Generated at mm/dd/yy hh:mm:ss EDT  
IPL at mm/dd/yy hh:mm:ss EDT
```

Determine the GCS Service Level

In the example, 001 represents the RSU service level (0001).

```
query gcslevel  
z/VM version 5 release 3, Service Level 001
```

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 all 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, New York 10504-1785
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 106-0032, 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
2455 South Road
Poughkeepsie, New York 12601-5400
U.S.A.
Attention: Information Request

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 document 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 illustrate 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

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

C/VM	Performance Toolkit for VM
DFSMS/VM	RACF
ECKD	System z
eServer	System z9
IBM	VM/ESA
IBMLink	VTAM
Language Environment	z/OS
MVS	z/VM
OpenExtensions	zSeries

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Glossary

For a list of z/VM terms and their definitions, see the *z/VM: Glossary* book.

The glossary is also available through the online HELP Facility. For example, to display the definition of “cms”, enter:

```
help glossary cms
```

You will enter the glossary HELP file and the definition of “cms” will be displayed as the current line. While you are in the glossary HELP 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 the *z/VM: CMS User's Guide*.

Bibliography

This bibliography lists the publications in the IBM® z/VM® product library, and it may list other publications referenced in this publication. For abstracts of z/VM publications and information about current editions and available media, see *z/VM: General Information*.

Where to Get z/VM Publications

z/VM publications are available from the following sources:

- IBM Publications Center at www.elink.ibm.com/publications/servlet/pbi.wss
- z/VM Internet Library at www.ibm.com/eserver/zseries/zvm/library/
- *IBM Online Library: z/VM Collection* (CD-ROM), SK2T-2067
- *IBM Online Library: z/VM Collection on DVD*, SK5T-7054

z/VM Base Library

Overview

- *z/VM: General Information*, GC24-6095
- *z/VM: Glossary*, GC24-6097
- *z/VM: License Information*, GC24-6102

Installation, Migration, and Service

- *z/VM: Guide for Automated Installation and Service*, GC24-6099
- *z/VM: Migration Guide*, GC24-6103
- *z/VM: Service Guide*, GC24-6117
- *z/VM Summary for Automated Installation and Service (DVD Installation)*, GA76-0406
- *z/VM Summary for Automated Installation and Service (Tape Installation)*, GA76-0407
- *z/VM: VMSES/E Introduction and Reference*, GC24-6130

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6074
- *z/VM: CMS Planning and Administration*, SC24-6078
- *z/VM: Connectivity*, SC24-6080

- *z/VM: CP Planning and Administration*, SC24-6083
- *z/VM: Getting Started with Linux on System z*, SC24-6096
- *z/VM: Group Control System*, SC24-6098
- *z/VM: I/O Configuration*, SC24-6100
- *z/VM: Running Guest Operating Systems*, SC24-6115
- *z/VM: Saved Segments Planning and Administration*, SC24-6116
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6140
- *z/VM: TCP/IP Planning and Customization*, SC24-6125
- *eServer zSeries 900: Planning for the Open Systems Adapter-2 Feature*, GA22-7477
- *System z9 and eServer zSeries: Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7935
- *System z9 and eServer zSeries 890 and 990: Open Systems Adapter-Express Integrated Console Controller User's Guide*, SA22-7990
- *z/OS and z/VM: Hardware Configuration Manager User's Guide*, SC33-7989

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6082
- *z/VM: Performance*, SC24-6109

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6073
- *z/VM: CMS Pipelines Reference*, SC24-6076
- *z/VM: CMS Pipelines User's Guide*, SC24-6077
- *z/VM: CMS Primer*, SC24-6137
- *z/VM: CMS User's Guide*, SC24-6079
- *z/VM: CP Commands and Utilities Reference*, SC24-6081
- *z/VM: System Operation*, SC24-6121
- *z/VM: TCP/IP User's Guide*, SC24-6127
- *z/VM: Virtual Machine Operation*, SC24-6128
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6131
- *z/VM: XEDIT User's Guide*, SC24-6132
- *CMS/TSO Pipelines Author's Edition*, SL26-0018

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6069
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6070
- *z/VM: CMS Application Multitasking*, SC24-6071
- *z/VM: CMS Callable Services Reference*, SC24-6072
- *z/VM: CMS Macros and Functions Reference*, SC24-6075
- *z/VM: CP Programming Services*, SC24-6084
- *z/VM: CPI Communications User's Guide*, SC24-6085
- *z/VM: Enterprise Systems Architecture/Extended Configuration Principles of Operation*, SC24-6094
- *z/VM: Language Environment User's Guide*, SC24-6101
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6104
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6105
- *z/VM: OpenExtensions Commands Reference*, SC24-6106
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6107
- *z/VM: OpenExtensions User's Guide*, SC24-6108
- *z/VM: Program Management Binder for CMS*, SC24-6110
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6112
- *z/VM: REXX/VM Reference*, SC24-6113
- *z/VM: REXX/VM User's Guide*, SC24-6114
- *z/VM: Systems Management Application Programming*, SC24-6122
- *z/VM: TCP/IP Programmer's Reference*, SC24-6126
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- *z/OS: Language Environment Concepts Guide*, SA22-7567
- *z/OS: Language Environment Debugging Guide*, GA22-7560
- *z/OS: Language Environment Programming Guide*, SA22-7561

- *z/OS: Language Environment Programming Reference*, SA22-7562
- *z/OS: Language Environment Run-Time Messages*, SA22-7566
- *z/OS: Language Environment Writing ILC Applications*, SA22-7563
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *z/OS MVS Program Management: User's Guide and Reference*, SA22-7643

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6118
- *z/VM: CP Messages and Codes*, GC24-6119
- *z/VM: Diagnosis Guide*, GC24-6092
- *z/VM: Dump Viewing Facility*, GC24-6093
- *z/VM: Other Components Messages and Codes*, GC24-6120
- *z/VM: TCP/IP Diagnosis Guide*, GC24-6123
- *z/VM: TCP/IP Messages and Codes*, GC24-6124
- *z/VM: VM Dump Tool*, GC24-6129
- *z/OS and z/VM: Hardware Configuration Definition Messages*, SC33-7986

Publications for z/VM Optional Features

Data Facility Storage Management Subsystem for VM

- *z/VM: DFSMS/VM Customization*, SC24-6086
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6087
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6088
- *z/VM: DFSMS/VM Planning Guide*, SC24-6089
- *z/VM: DFSMS/VM Removable Media Services*, SC24-6090
- *z/VM: DFSMS/VM Storage Administration*, SC24-6091

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6133
- *z/VM: Directory Maintenance Facility Messages*, GC24-6134

- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6135

Performance Toolkit for VM™

- *z/VM: Performance Toolkit Guide*, SC24-6156
- *z/VM: Performance Toolkit Reference*, SC24-6157

RACF® Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6143
- *z/VM: RACF Security Server Command Language Reference*, SC24-6144
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6145
- *z/VM: RACF Security Server General User's Guide*, SC24-6146
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6147
- *z/VM: RACF Security Server Messages and Codes*, GC24-6148
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6142
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6149
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6150

RSCS Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6151
- *z/VM: RSCS Networking Exit Customization*, SC24-6152
- *z/VM: RSCS Networking Messages and Codes*, GC24-6153
- *z/VM: RSCS Networking Operation and Use*, SC24-6154
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6155

Index

Special characters

- \$source files
 - \$DASD\$ \$CONSTS 6-10
 - AVS 6-9
 - CMS component 6-9
 - CP component 6-9
 - Dump Viewing Facility component 6-9
 - example with DMSSBINS E-3
 - exceptions 6-10
 - GCS 6-9
 - overview 6-9
 - replacement part 6-9, 7-3
 - replacement part REXX files 6-8
 - REXX/VM component 6-9
 - TSAF 6-9
 - updating 6-9, 7-2
 - updating REXX files 6-8
 - VMSES/E component 6-9

A

- AGWBLHLP EXEC J-7
- AGWBLINF EXEC J-7
- AGWBLLIB EXEC J-7
- AGWBLSRC EXEC J-7
- AGWBLUCE EXEC J-7
- AGWBLUTL EXEC J-7
- alternate
 - apply disk 1-7
 - disk 1-2
- AMQOM MACLIB J-3
- APAR descriptions 1-7
- appid VVT\$PSU\$ 2-4
- application of service, flowchart 3-2
- APPLY disks 1-7
- applying service
 - AVS component 3-63
 - CMS component 3-27
 - CP component 3-36
 - Dump Viewing Facility component 3-51
 - GCS component 3-44
 - Language Environment component 3-21
 - local service 5-3
 - overview 1-3
 - REXX/VM component 3-15
 - TSAF component 3-57
 - VMSES/E component 3-9
- ASDREUSP MACLIB J-3
- ATSBLHLP EXEC J-7
- ATSBLINF EXEC J-7
- ATSBLMES EXEC J-7
- ATSBLSRC EXEC J-7
- ATSBLUCE EXEC J-7
- ATSMLOAD EXEC J-7
- audience of this book xi
- AUX
 - default AUX identifier 5-5

AUX (*continued*)

- files 1-4

AVS

- \$source files 6-9
- build lists
 - all 4-38
 - description J-1
 - for select and copy functions 4-38
- Build Status Table, updating 3-64
- building new objects 3-64
- detailed build tasks
 - build preparation 4-38
 - determining build lists to process 4-38
 - load libraries, rebuilding 4-39
 - select and copy serviced files 4-38
- load libraries 4-39
- merge processing 3-65
- service disks and directories 4-38
- service procedure 3-61

B

- back-up copy, primary parm disk 3-39, 4-26
- BASE disks 1-8
- build
 - lists J-1
 - tasks by component 1-14
 - the CMS Nucleus 3-17, 3-31, 3-42, 4-5, 4-19
 - the CP Nucleus 3-38, 4-25
 - the GCS Nucleus 3-17, 3-42, 3-47, 4-5, 4-30
- BUILD disks 1-8
- build lists with VMSES/E
 - AGWBLHLP 4-38
 - AGWBLINF 4-38
 - AGWBLLIB 4-38
 - AGWBLSRC 4-38
 - AGWBLUCE 4-38
 - AGWBLUTL 4-38
 - AGWSBHLP 4-38
 - AMQOM 4-13
 - ATSBLHLP 4-36
 - ATSBLINF 4-36
 - ATSBLMES 4-36
 - ATSBLSRC 4-36
 - ATSBLUCE 4-36
 - ATSMLOAD 4-36
 - ATSSBHLP 4-36
 - AVS 4-38
 - CDDBLSAM 4-7
 - CEEBLFPR 4-7
 - CEEBLKLD 4-7
 - CEEBLMAC 4-7
 - CEEBLMOD 4-7
 - CEEBLNLS 4-7
 - CEEBLOBJ 4-7
 - CEEBLRUN 4-7
 - CEEBLSB2 4-7
 - CEEBLSGA 4-7

build lists with VMSES/E *(continued)*

CEEBLSGB 4-7
 CMS 4-12
 CMSALOAD 4-13
 CMSLOAD 4-13
 CMSMAA 4-13
 CMSMLOAD 4-13
 CMSXLOAD 4-13
 CP 4-22
 CPLOAD 4-23
 DMSBL400 4-13
 DMSBL490 4-13
 DMSBL493 4-13
 DMSBL990 4-13
 DMSBLAEN 4-13
 DMSBLAMT 4-13
 DMSBLASM 4-13
 DMSBLBKW 4-13
 DMSBLCEN 4-13
 DMSBLCMT 4-13
 DMSBLDOS 4-13
 DMSBLFAS 4-13
 DMSBLHLP 4-13
 DMSBLINF 4-13
 DMSBLNUH 4-13
 DMSBLPRP 4-13
 DMSBLRAS 4-13
 DMSBLSRC 4-13
 DMSBLVBL 4-13
 DMSBLVML 4-13
 DMSBLVMT 4-13
 DMSGPI 4-13
 DMSOM 4-13
 DMSSBAME 4-13
 DMSSBBAM 4-13
 DMSSBDOS 4-13
 DMSSBGER 4-13
 DMSSBHLP 4-13
 DMSSBINS 4-13
 DMSSBKAN 4-13
 DMSSBPIP 4-13
 DMSSBSFS 4-13
 DMSSBSVM 4-13
 DMSSBUCE 4-13
 DMSSBVML 4-13
 Dump Viewing Facility 4-33
 EDCBL49B 4-7
 EDCBLCPP 4-7
 EDCBLFPR 4-7
 EDCBLH 4-7
 EDCBLHPRE 4-7
 EDCBLHPU 4-7
 EDCBLLKD 4-7
 EDCBLMAC 4-7
 EDCBLMOD 4-7
 EDCBLPMB 4-7
 EDCBLRUN 4-7
 EDCBSB2 4-7
 EDCBLSP1 4-7
 EDCBLSP2 4-7
 EDCSBHLP 4-7

build lists with VMSES/E *(continued)*

GCS 4-28
 GCTBLHLP 4-28
 GCTBLINF 4-28
 GCTBLPRP 4-28
 GCTBLRAS 4-28
 GCTBLRST 4-28
 GCTGPI 4-28
 GCTLOAD 4-28
 GCTSBHLP 4-28
 GSUBLLID 4-13
 GSUBLSYS 4-13
 GSUMLOAD 4-13
 GUIBLCSL 4-13
 GUISB CSL 4-13
 GUISBDSK 4-13
 HCPBL490 4-23
 HCPBL493 4-23
 HCPBLHLP 4-23
 HCPBLIMG 4-23
 HCPBLINS 4-23
 HCPBLNUC 4-23
 HCPBLRAS 4-23
 HCPBLSAM 4-23
 HCPBLSDK 4-23
 HCPBLSRC 4-23
 HCPBLUCE 4-23
 HCPBLUTL 4-23
 HCPGPI 4-23
 HCPMLOAD 4-23
 HCPOM1 4-23
 HCPOM2 4-23
 HCPSPI 4-23
 HCPSBAME 4-23
 HCPSBGER 4-23
 HCPSBHLP 4-23
 HCPSBKAN 4-23
 HCPSBUCE 4-23
 HCPXLOAD 4-23
 HCQMLOAD 4-23
 HCSBL493 4-33
 HCSBLHLP 4-33
 HCSBLMSG 4-33
 HCSBLSRC 4-33
 HCSBLUCE 4-33
 HCSMLOAD 4-33
 HCSSBHLP 4-33
 HCSSLLOAD 4-33
 IBMBLCAL 4-7
 IBMBLFPR 4-7
 IBMBLLKD 4-7
 IBMBLMOD 4-7
 IBMBLMTH 4-7
 IBMBLNLS 4-7
 IBMBLRUN 4-7
 IBMBLSAM 4-7
 IGXBFLPR 4-7
 IGZBLLBO 4-7
 IGZBLLKD 4-7
 IGZBLMAC 4-7
 IGZBLMOD 4-7

build lists with VMSES/E *(continued)*

- IGZBLNLS 4-7
- IGZBLPAC 4-7
- IGZBLRUN 4-7
- IGZBLSAM 4-7
- IOCPLOAD 4-13
- IXXBLSNUC 4-4
- IXXBLSRC 4-4
- IXXOM 4-4
- Language Environment 4-6
- MVSXA 4-13
- OSMACRO 4-13
- OSMACRO1 4-13
- OSPSI 4-13
- REXX/VM 4-4
- SERVLOAD 4-13
- TSAF 4-36
- VMFBLHLP 4-1
- VMFBLNON 4-1
- VMFBLSES 4-1
- VMFBLSRC 4-1
- VMFBLSYS 4-1
- VMFMLOAD 4-1
- VMFSBHLP 4-1
- VMFSLOAD 4-1
- VMSES/E 4-1

building new levels of service

- AVS component 3-64
- CMS component 3-28
- CP component 3-37
- disks accessed 1-4
- Dump Viewing Facility component 3-52
- GCS component 3-45
- individual objects 4-1
- Language Environment component 3-21
- overview 1-4, 1-13
- REXX/VM component 3-16
- TSAF component 3-57
- VMSES/E component 3-10

C

- CEEBLFPR J-1
- CEEBLKLD 4-8, J-1
- CEEBLMAC J-1
- CEEBLMOD J-1
- CEEBLNLS J-1
- CEEBLOBJ 4-8, J-1
- CEEBLRUN J-1
- CEEBLSAM J-1
- CEEBLSB2 4-8, J-2
- CEEBLSGA J-2
- CEEBLSGB J-2
- CF1 parm disk 3-74
- CF2 parm disk 3-74
- changing IBM service defaults 2-6, 3-5
- changing, CP load list F-1
- CMS
 - \$source files 6-9
 - build lists
 - all 4-12

CMS *(continued)*

build lists *(continued)*

- and MODULEs to be rebuilt 4-17
- description J-1
- for rebuilding executable MODULEs 4-16
- for select and copy functions 4-13

Build Status Table, updating 3-28

building new objects 3-28

detailed build tasks

- Assemble module, rebuilding 4-17
- build preparation 4-12
- Callable Services Libraries, rebuilding 4-17
- CSL, rebuilding 4-17
- determining build lists to process 4-12
- DMSBLFAS, rebuilding 4-17
- DMSBLPRP, rebuilding 4-15
- executable MODULEs, rebuilding 4-16
- IPL the CMS nucleus 4-19
- load library, rebuilding 4-15
- MACLIBs, updating 4-14
- rebuild the CMS nucleus 4-19
- select and copy serviced files 4-13
- TXTLIBs, rebuilding 4-15

determine CMS service level K-1

IPL the nucleus 3-31

MACLIBs 4-14

merge processing 3-32

service disks 4-12

service disks and directories 4-12

service procedure 3-25

testing the new level D-1

CMS nucleus

rebuilding 3-31, 4-19

CMS nucleus, IPLing 3-31

CMS saved system 3-73

CMS/DOS saved segments 3-69

CMSALOAD MACLIB J-3

CMSCLOAD MACLIB J-3

CMSFILES saved segment 3-69

CMSINST

CMSINST LSEG E-1

DMSSBINS EXEC E-1

local modification E-1

rebuilding 3-70

updating E-1

CMSINST saved segment 3-69

CMSLOAD EXEC J-3

CMSMLOAD EXEC J-3

CMSPIPES saved segment 3-69, 3-70

CMSSAA EXEC J-3

CMSSAA TXTLIB 4-15

CMSVMLIB saved segment 3-69

CMSVSAM segment 3-29

CMSXLOAD EXEC J-3

commands used for service 1-14

component(s)

build dependencies 1-4

names 1-6

of z/VM 1-2

order of service 1-2, 1-5

console spooling 2-4, 3-3

- COR service
 - flowchart of application 3-2
 - main tasks 1-2
 - method of installing 3-1
 - order of application 1-2
 - overview 1-2
 - preparing
 - disks 2-6, 3-5
 - high common end storage 2-6
 - the system 2-6, 3-5
 - procedure 3-1
 - procedure overview 3-1
 - receiving service documentation 3-6
 - substituting disks 3-3
 - using substitute disks 3-3
- corequisite publications xi, 1-1
- CP
 - \$source files 6-9
 - build lists
 - all 4-22
 - description J-1
 - for CP utilities 4-25
 - for rebuilding executable MODULEs 4-24
 - for select and copy functions 4-23
 - Build Status Table, updating 3-37
 - building new objects 3-37
 - detailed build tasks
 - build preparation 4-22
 - CP nucleus build preparation 4-25
 - determining build lists to process 4-22
 - executable MODULEs, rebuilding 4-24
 - MACLIBs, updating 4-23
 - rebuild the CP nucleus 4-25
 - rebuilding CP utilities 4-25
 - save the CP load map 4-25
 - select and copy serviced files 4-23
 - stand-alone dump, rebuilding 4-25
 - determine CP service level K-1
 - MACLIBs 4-23
 - merge processing 3-40
 - service disks 4-22
 - service disks and directories 4-22
 - service procedure 3-34
 - stand-alone dump, rebuilding 3-38
- CP load list, updating F-1, F-3
- CP load map 3-39
- CP nucleus, rebuilding 4-25
- CP nucleus, secondary parm disk 3-40, 4-26
- CPLOAD EXEC J-5
- CPRELEASE 3-74
- CSL (Callable Services Library) 4-17
- customer local service 5-2

D

- DDR CMS system disk 3-72
- Decide
 - to use preapplied prebuilt service 2-7
- defaults for service, changing 2-6, 3-5
- DELTA disk 1-7

- detailed build steps
 - AVS 4-38
 - CMS 4-12
 - CP 4-22
 - Dump Viewing Facility 4-33
 - GCS 4-28
 - how to use 4-1
 - Language Environment 4-6
 - overview 4-1
 - REXX/VM 4-4
 - TSAF 4-36
 - VMSES/E 4-1
- determine CMS service level K-1
- determine CP service level K-1
- determine GCS service level K-1
- determine RSU service level 3-1
- DIRECTXA
 - tool 1-14
- disk preparation for COR service 3-5
- disk(s)
 - used for servicing
 - AVS component 1-12
 - CMS component 1-9
 - CP component 1-10
 - Dump Viewing Facility component 1-10
 - GCS component 1-11
 - Language Environment component 1-13
 - REXX/VM component 1-9
 - TSAF component 1-12
 - VMSES/E component 1-8
 - z/VM 1-3
- disks used to service z/VM 1-7
- DMSBL400 EXEC J-3
- DMSBL490 EXEC J-3
- DMSBL493 EXEC J-3
- DMSBL990 EXEC J-3
- DMSBLAEN EXEC J-3
- DMSBLAEN TXTLIB 4-15
- DMSBLAMT EXEC J-3
- DMSBLAMT TXTLIB 4-15
- DMSBLASM EXEC J-4
- DMSBLBKW EXEC J-4
- DMSBLBKW TXTLIB 4-15
- DMSBLCEN EXEC J-4
- DMSBLCEN TXTLIB 4-15
- DMSBLCMT EXEC J-4
- DMSBLCMT TXTLIB 4-15
- DMSBLCRX EXEC J-4
- DMSBLDOS EXEC J-4
- DMSBLFAS EXEC J-4
- DMSBLHLP EXEC J-4
- DMSBLINF EXEC J-4
- DMSBLNUH EXEC J-4
- DMSBLPRP EXEC J-4
- DMSBLRAS EXEC J-4
- DMSBLSRC EXEC J-4
- DMSBLUCE EXEC J-4
- DMSBLVBL EXEC J-4
- DMSBLVML EXEC J-4
- DMSBLVMT EXEC J-4
- DMSGPI EXEC J-4

- DMSNGP 1-7
- DMSOM EXEC J-4
- DMSRP MACLIB J-4
- DMSSBAME EXEC J-4
- DMSSBVML EXEC J-5
- DOSINST saved segment 3-69
- Dump Viewing Facility
 - \$source files 6-9
 - build lists
 - all 4-33
 - description J-1
 - for rebuilding executable MODULEs 4-34
 - for select and copy functions 4-33
 - Build Status Table, updating 3-52
 - building new objects 3-52
 - detailed build tasks
 - build preparation 4-33
 - determining build lists to process 4-33
 - executable MODULEs, rebuilding 4-34
 - select and copy serviced files 4-33
 - merge processing 3-53
 - service disks 4-33
 - service procedure 3-49

E

- EDCBL49B J-2
- EDCBLCPP 4-8, J-2
- EDCBLFPR J-2
- EDCBLH J-2
- EDCBLHPE J-2
- EDCBLHPU J-2
- EDCBLLKD 4-8, J-2
- EDCBLMAC J-2
- EDCBLMOD J-2
- EDCBLPMB J-2
- EDCBLRUN J-2
- EDCBLSB2 4-8, J-2
- EDCBLSP1 4-8, J-2
- EDCBLSP2 4-8, J-2
- EDCSBHLP J-2
- emergency service 5-1
- envelope
 - non-VMSES B-1
 - VMSES A-1
- examples, layout of xi

F

- file type abbreviation
 - obtaining 5-5
- files
 - \$\$\$TLL\$\$, CMS example 3-31
 - \$VMFAPP \$MSGLOG, CMS example 3-27
 - \$VMFBLD \$MSGLOG, CMS example 3-30
 - \$VMFINS \$MSGLOG 2-9
 - \$VMFMRD \$MSGLOG, CMS example 3-33
 - appid VVT\$PSU\$ 2-4
 - CMSLOAD G-1
 - COR DOCUMENT A-2
 - COR VMELEDOC A-2, B-1

files (*continued*)

- COR01 MEDIAMAP B-1
- default apply identifier 5-5
- default AUX identifier 5-5
- default for control 5-5
- default VVT identifier 5-5
- DMSPARMS D-1
- GCTLOAD G-1
- nsdocenvfn SERVLINK B-1
- override 2-6
- ppfname 2-6
- PSUPLAN 2-7
- SERVICE DOCUMENT B-1
- SERVICE SERVLINK A-2
- TXTLIBs 4-8, 4-15
- VLSTnnnn SERVLINK A-1
- flowchart of service application 3-2
- format of this book xi
- FPLGPI MACLIB J-5
- FPLOM MACLIB J-5

G

- GCS
 - \$source files 6-9
 - build lists
 - all 4-28
 - description J-1
 - for select and copy functions 4-28
 - Build Status Table, updating 3-45
 - building new objects 3-45
 - detailed build tasks
 - build preparation 4-28
 - determining build lists to process 4-28
 - GCS load library, rebuilding 4-29
 - GCS nucleus, rebuilding 4-30
 - IPL the GCS nucleus 4-30
 - MACLIBs, updating 4-29
 - rebuild the GCS nucleus 4-30
 - select and copy serviced files 4-28
 - determine GCS service level K-1
 - MACLIBs 4-29
 - merge processing 3-48
 - nucleus, IPLing 3-47
 - service disks and directories 4-28, 4-33
 - service procedure 3-42
 - GCS nucleus, IPLing 4-30
 - GCS nucleus, rebuilding 4-30
 - GCTBLHLP EXEC J-6
 - GCTBLINF EXEC J-6
 - GCTBLPRP EXEC J-6
 - GCTBLRAS EXEC J-6
 - GCTBLRST EXEC J-6
 - GCTBLUCE EXEC J-6
 - GCTGPI EXEC J-6
 - GCTLOAD EXEC J-6
 - glossary information M-1
 - GSUBLLIB EXEC J-5
 - GSUBLLIB TXTLIB 4-15
 - GSUBLSYS EXEC J-5
 - GSUMLOAD EXEC J-5

GUIBLCSL EXEC J-5
 GUICSLIB 3-70
 GUICSLIB saved segment 3-70
 GUIBCSL EXEC J-5

H

HCPBL490 EXEC J-5
 HCPBL493 EXEC J-5
 HCPBLHLP EXEC J-5
 HCPBLIMG EXEC J-5
 HCPBLINS EXEC J-5
 HCPBLRAS EXEC J-5
 HCPBLSAM EXEC J-5
 HCPBLSDK EXEC J-5
 HCPBLSRC EXEC J-6
 HCPBLUCE EXEC J-6
 HCPBLUTL EXEC J-6
 HCPESAME EXEC J-6
 HCPGPI EXEC J-6
 HCPMDLAT MACRO
 CP load list F-1
 updating xxxMDLAT MACRO F-1
 HCPMLOAD EXEC J-6
 HCPOM1 EXEC J-6
 HCPOM2 EXEC J-6
 HCPPSI EXEC J-6
 HCPRIO 1-7
 HCPRP MACLIB J-6
 HCPSBAME EXEC J-6
 HCPSYS 1-7
 HCPXLOAD EXEC J-6
 HCSBL493 EXEC J-7
 HCSBLHLP EXEC J-7
 HCSBLMSG EXEC J-7
 HCSBLSRC EXEC J-7
 HCSBLUCE EXEC J-7
 HCSSLLOAD EXEC J-7
 HCSSLLOAD EXEC J-7
 HELP saved segment 3-70
 example, CMSINST E-1
 HELP VMSES MENU xii
 HELP, online 1-17
 HELP, using online 1-17
 HELPSEG 3-70
 how to use HELP xii

I

IBM local service 5-1
 IBM service defaults, changing 2-6, 3-5
 IBMBLCAL 4-8, J-2
 IBMBLFPR J-2
 IBMBLLKD 4-8, J-2
 IBMBLMOD J-2
 IBMBLMTH 4-8, J-2
 IBMBLNLS J-3
 IBMBLRUN J-3
 IBMBLSAM J-3
 IGZBLFPR J-3
 IGZBLLBO J-3

IGZBLLKD 4-8, J-3
 IGZBLMAC J-3
 IGZBLMOD J-3
 IGZBLNLS J-3
 IGZBLPAC J-3
 IGZBLRUN J-3
 IGZBLSAM J-3
 installing preventive maintenance 1-1
 intermediate
 apply disk 1-7
 disk 1-2
 IOCPLOAD EXEC J-5
 IPL the CMS nucleus 4-19
 IPL the GCS nucleus 4-30
 IPL the Nucleus
 the CMS Nucleus 3-31, 4-21
 the GCS Nucleus 3-47, 4-31
 IXXBLNUC EXEC J-1
 IXXBLSRC EXEC J-1
 IXXOM EXEC J-1
 IXXRP MACLIB J-1

L

Language Environment
 build lists
 all 4-6
 description J-1
 for rebuilding executable MODULEs 4-9
 for select and copy functions 4-7
 Build Status Table, updating 3-22
 building new objects 3-21
 detailed build tasks
 build preparation 4-6
 CEEBLRUN, rebuilding 4-9
 determining build lists to process 4-6
 EDCBLRUN, rebuilding 4-9
 executable MODULEs, rebuilding 4-9
 load library, rebuilding 4-9
 MACLIBs, updating 4-8
 select and copy serviced files 4-7
 TXTLIBs, rebuilding 4-8
 MACLIBs 4-8
 merge processing 3-23
 service disks and directories 1-13
 service procedure 3-19
 load list, updating CP load list F-1
 load map, CP 3-39
 local
 disk 1-7
 example, CMSINST E-1
 sample disk 1-7
 Version Vector Table 6-5, H-2
 local service
 \$source files, updating 6-9, 7-2
 \$source files, updating, compiled REXX files 6-8
 add update to AUX file 6-4
 applying 5-3
 ASSEMBLE files 6-5, 6-7
 CP load list F-1
 CPLOAD EXEC F-1

local service *(continued)*

- create replacement part 6-6, 6-9
 - create replacement part compiled REXX files 6-8
 - create update 6-4
 - customer 5-2
 - definition 5-1
 - definitions 5-3
 - disks used 6-2
 - DLCS file type, rebuilding 6-12
 - DMSTRTxx file, rebuilding 6-12
 - example, CMSINST E-1
 - guidelines for update files 6-3
 - IBM 5-1
 - MACLIBs to be rebuilt 7-6
 - MACLIBs, updating 6-5
 - MACROs 6-7
 - message repository, rebuilding 6-12
 - NLS files, compiling 6-12, 7-8
 - overview 1-4, 5-1
 - patches, applying and removing I-1
 - preparation 5-1, 6-1
 - re-applying
 - AVS component 3-63
 - CMS component 3-27
 - CP component 3-36
 - Dump Viewing Facility component 3-51
 - GCS component 3-44
 - Language Environment component 3-21
 - REXX/VM component 3-15
 - TSAF component 3-57
 - VMSES/E component 3-9
 - rebuild objects 6-7
 - ASSEMBLE files 6-10
 - DLCS files 6-12
 - DMSTRTxx file 6-12
 - MACLIBs 6-10
 - message repository 6-12
 - NLS files 6-12
 - remaining objects 6-12
 - replaced ASSEMBLE files 6-11
 - REPOS files 6-12
 - source files 6-8
 - receiving 6-2
 - replaced ASSEMBLE files 6-11
 - REPOS file type, rebuilding 6-12
 - reworking overview 5-2
 - reworking procedures 7-1
 - source updates 6-3
 - vendor 5-2
 - Version Vector Table, updating 6-5, H-2
 - VVT, updating 6-5, H-2
 - zaps, applying H-1
- LOCALMOD disk 1-7
- LOCALSAM disk 1-7
- LOGO CONFIG 1-7

M

- merge processing
 - AVS 3-65
 - CMS 3-32

merge processing *(continued)*

- CP 3-40
- Dump Viewing Facility 3-53
- GCS 3-48
- Language Environment 3-23
- overview 1-3
- REXX/VM 3-18
- TSAF 3-59
- VMSES/E 3-11
- Move files into the byte file system 3-71
- MVSXA J-5

N

- national languages, changing C-1
- NLS files, compiling 6-12, 7-8
- NLS, changing defaults C-1
- non-VMSES envelopes, receiving B-1
- nucleus
 - changing the national language C-1
 - IPL the CMS nucleus 4-19
 - IPL the GCS nucleus 4-30
 - rebuild the CMS nucleus 4-19
 - rebuild the CP nucleus 4-25
 - rebuild the GCS nucleus 4-30
 - save the CP load map 4-25

O

- object, building G-1
- objects 1-8
- online HELP Facility, using 1-17
- order of service 1-5
- OSMACRO EXEC J-5
- OSMACRO1 EXEC J-5
- OSPFI EXEC J-5
- out-of-component requisites 1-6
- overview of procedures 1-1

P

- packaging service 1-6
- patches,
 - applying and removing I-1
 - creating I-1
- place into production 3-72
- PPF (Product Parameter File)
 - base 1-6
 - base file names 1-6
 - override 1-6
 - override file names 1-6
 - service packaging 1-6
- Preapplied Prebuilt Service
 - Preferred Method 2-7
 - RSU, how to decide 2-7
- preparation for service
 - AVS component 3-61
 - CMS component 3-25
 - CP component 3-34
 - Dump Viewing Facility component 3-49
 - GCS component 3-42

- preparation for service *(continued)*
 - Language Environment component 3-19
 - local modification 6-1
 - local service 6-1
 - REXX/VM component 3-13
 - TSAF component 3-55
 - VMSES/E component 3-7
- prerequisites to using this book xi
- preventive maintenance 1-1
- primary parm disk, back-up copy 3-39, 4-26
- procedures, layout of xi
- procedures, overview 1-1
- production
 - apply disk 1-8
 - delta disk 1-7
 - disk 1-2
 - level system 3-67
- production, placing serviced components into
 - AVS 3-75
 - CMS 3-73
 - CMS saved system 3-73
 - CMS/DOS saved segment 3-69
 - CMSFILES saved segment 3-69
 - CMSINST saved segment 3-69
 - CMSPIPES saved segment 3-69, 3-70
 - CMSVMLIB saved segment 3-69
 - CP nucleus 3-73
 - DOSINST saved segment 3-69
 - GCS 3-75
 - GUICSLIB saved segment 3-70
 - HELP saved segment 3-70
 - merging tested service 3-75
 - overview 1-4, 3-67
 - saved segments, rebuilding 3-67
 - SVM saved segment 3-69
 - test CMS system disk 3-72
 - test tools disk 3-72
 - TSAF 3-75
 - VMLIB logical segment 3-69
- PSP (Preventive Service Planning)
 - processing 2-11
- PSU (Product Service Upgrade)
 - build new service level 2-11
 - DASD sizes, determining 2-5
 - documentation needed 2-3
 - effect on tailored files 2-1
 - intended use 1-1
 - limitations 1-1
 - materials needed for use 2-3
 - overview 1-1
 - planning for use 2-1
 - preparation for 2-4
 - process additional service 2-9
 - reach-ahead service, determining 2-7
 - receiving the refresh tape 2-7
 - rework local service 2-10
 - using 2-1
- PTF-numbered parts 1-7
- PTFs Only
 - Alternate Method 2-7
 - RSU, how to decide 2-7

- purpose of this book xi
- PUT service procedure 3-1

R

- Rebuild the CMS nucleus 4-19
- Rebuild the CP nucleus 4-25
- Rebuild the GCS nucleus 4-30
- Rebuild the Nucleus
 - the CMS Nucleus 3-17, 3-31, 3-42, 4-5, 4-19
 - the CP Nucleus 3-38, 4-25
 - the GCS Nucleus 3-17, 3-42, 3-47, 4-5, 4-30
- Rebuild the saved segments 3-67
- receiving service
 - AVS component 3-62
 - CMS component 3-26
 - CP component 3-35
 - Dump Viewing Facility component 3-50
 - electronically A-1
 - GCS component 3-43
 - Language Environment component 3-20
 - local service 6-2
 - non-VMSES envelopes B-1
 - overview 1-3
 - REXX/VM component 3-14
 - TSAF component 3-56
 - using ServiceLink A-1
 - VMSES/E component 3-8
- Recommended Service Upgrade 1-1
- reference service information 1-1
- removing service
 - backing out a local modification 8-3
 - backing out a single PTF 8-2
 - local modification 8-3
 - overview 8-1
 - service level 8-1
 - service levels 8-1
 - single PTF 8-2
- replacement part
 - create 6-9
- replacement part REXX files
 - create 6-8
 - VMFREPL EXEC 6-9
- rework local service 5-2
- reworking local service
 - procedures 7-1
 - replacement-maintained 7-3
 - source-maintained 7-2
- REXX/VM
 - \$source files 6-9
 - build lists
 - all 4-4
 - description J-1
 - for select and copy functions 4-4
 - Build Status Table, updating 3-16
 - building new objects 3-16
 - detailed build tasks
 - build preparation 4-4
 - determining build lists to process 4-4
 - MACLIBs, updating 4-5
 - rebuild the nuclei 4-5

- REXX/VM *(continued)*
 - detailed build tasks *(continued)*
 - select and copy serviced files 4-4
 - MACLIB 4-5
 - merge processing 3-18
 - rebuild the nuclei 3-17
 - service disks 4-4
 - service disks and directories 4-4
 - service procedure 3-13
- RSU (Recommended Service Upgrade)
 - detailed overview 2-2
 - determine RSU service level 3-1
 - for use with preventive service 2-1
 - overview 1-1
 - relationship to PSU 2-1
 - RSU files 2-1
 - service application flowchart 2-2

S

- SA.DUMP, rebuilding 3-38
- Save the CP load map 4-25
- saved segments
 - CMS/DOS 3-69
 - CMSFILES 3-69
 - CMSINST 3-69
 - CMSPIPES 3-69, 3-70
 - CMSVMLIB 3-69
 - DOSINST 3-69
 - GUICSLIB 3-70
 - HELP 3-70
 - SVM 3-69
 - VSE/VSAM 3-29
- second level IPL 3-74
- secondary disk, CP nucleus 3-40, 4-26
- segments
 - CMS/DOS 3-69
 - CMSFILES 3-69
 - CMSINST 3-69
 - CMSPIPES 3-69, 3-70
 - CMSVMLIB 3-69
 - DOSINST 3-69
 - GUICSLIB 3-70
 - HELP 3-70
 - rebuilding 3-67
 - SVM 3-69
 - VSE/VSAM 3-29
- sequence of servicing components 1-5
- service
 - application flowchart 3-2
 - defaults, changing 2-6, 3-5
 - dependencies 1-5
 - disks 1-2
 - disks used 1-7
 - documentation, receiving 3-6
 - main tasks 1-2
 - non-VMSES envelopes B-1
 - packaging 1-6
 - procedure
 - AVS component 3-61
 - CMS component 3-25

- service *(continued)*
 - procedure *(continued)*
 - CP component 3-34
 - Dump Viewing Facility component 3-49
 - GCS component 3-42
 - Language Environment component 3-19
 - REXX/VM component 3-13
 - TSAF component 3-55
 - VMSES/E component 3-7
 - tools 1-14
 - VMSES envelopes A-1
- service level
 - CMS K-1
 - CP K-1
 - GCS K-1
 - RSU 3-1
- service procedures 1-1
- service refresh
 - build new service level 2-11
 - DASD sizes, determining 2-5
 - documentation needed 2-3
 - overview 2-1
 - preparation for 2-4
 - process additional service 2-9
 - reach-ahead service, determining 2-7
 - receiving the refresh tape 2-7
 - rework local service 2-10
- ServiceLink A-1
- SERVLOAD EXEC J-5
- source updates 6-3
- spooling your console 2-4, 3-3
- stand-alone dump 3-38
- SVM saved segment 3-69
- SYSTEM CONFIG 1-7
- SYSTEM disks 1-8
- system generation tools 1-14
- system preparation for COR service 2-6, 3-5

T

- TASK disks 1-7
- testing CMS D-1
- text libraries 4-8, 4-15
- tools used for service 1-14
- TSAF
 - \$source files 6-9
 - build lists
 - all 4-36
 - for select and copy functions 4-36
 - Build Status Table, updating 3-58
 - building new objects 3-57
 - detailed build tasks
 - build preparation 4-36
 - determining build lists to process 4-36
 - executable MODULEs, rebuilding 4-37
 - select and copy serviced files 4-36
 - merge processing 3-59
 - service disks and directories 4-36
 - service procedure 3-55
- two-column book format xi
- TXTLIBs 4-8, 4-15

U

- update files 6-3
- user entries, layout of xi
- using this book xi

V

- variables
 - aparnumber 5-3
 - appid 2-5, 5-3
 - auxft 5-3
 - cftabbrev 5-3
 - comment 5-3
 - compname 2-5, 5-3
 - ctlfile 5-3
 - curmodid 5-3
 - fm-local 5-3
 - fn 5-3
 - ft 5-3
 - ftabbrev 5-3
 - ftabbrev-modid 5-3
 - ftabbrev-ptfnumber 5-3
 - lcmodid 5-3
 - modid 5-3
 - opts 5-3
 - ppfname 2-6, 5-3
 - ptfnumber 5-3
 - realft 5-3
 - sftabbrev 5-3
 - svclvl 5-3
 - updateft 5-3
 - vvtft 5-3
- vendor service or updates 5-2
- VMFBLD
 - rebuild the CMS Nucleus 3-31
- VMFBLHLP EXEC J-1
- VMFBLNON EXEC J-1
- VMFBLSES EXEC J-1
- VMFBLSRC EXEC J-1
- VMFBLSYS EXEC J-1
- VMFBLUCE EXEC J-1
- VMFEXUPD command
 - usage notes 6-9
 - use for replacement parts 6-9, 7-5
 - using with DMSSBINS E-3
- VMFMLOAD EXEC J-1
- VMFMRDSK EXEC 1-3
- VMFPLCD command
 - using the command> B-1
- VMFPSU command
 - example output file 2-7
 - overview 2-3
 - using the command 2-7
- VMFREM EXEC 8-1
- VMFREPL EXEC 7-5
- VMFSBHLP EXEC J-1
- VMFSGMAP
 - example, CMSINST E-1
- VMFSLOAD EXEC J-1
- VMLIB D-1

VMSES/E

- \$source files 6-9
- build lists
 - all 4-1
 - description J-1
 - for processing HELP segment 4-3
 - for rebuilding executable MODULEs 4-2
 - for select and copy function 4-1
- Build Status Table, updating 3-10
- building new objects 3-10
- detailed build tasks
 - build preparation 4-1
 - determining build lists to process 4-1
 - executable MODULEs, rebuilding 4-2
 - select and copy serviced files 4-1
- merge processing 3-11
- service disks 4-1
- service disks and directories 4-1
- service procedure 3-7
- VMSES/E command information 1-1
- VSE/VSAM saved segment 3-29
- VVT
 - default VVT identifier 5-5
 - updating local 6-5, H-2

W

- who should read this book xi

X

- xxxMDLAT MACRO
 - CP load list F-1
 - updating xxxMDLAT MACRO F-1

Z

- z/VM HELP Facility, using 1-17
- z/VM Recommended Service Upgrade 1-1
- ZAP command H-1, I-1
- ZAPTEXT command H-1, I-1

Readers' Comments — We'd Like to Hear from You

z/VM
Service Guide
version 5 release 3

Publication No. GC24-6117-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your 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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



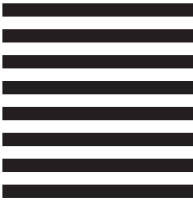
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, New York 12601-5400



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5741-A05

Printed in USA

GC24-6117-02



Spine information:



z/VM

Service Guide

version 5 release 3