

z/VM



Service Guide

Version 6 Release 3

Note:

Before using this information and the product it supports, read the information in "Notices" on page 51.

This edition applies to version 6, release 3, modification 0 of IBM z/VM (product number 5741-A07) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC24-6247-00.

© **Copyright IBM Corporation 1991, 2013.**

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

Contents

Figures	v
Tables	vii
About this document	ix
Intended audience	ix
Conventions and terminology	ix
Using the online HELP facility	x
Where to find more information	x
Links to other documents and web sites	x
How to send your comments to IBM.	xiii
Summary of Changes	xv
GC24-6247-01, z/VM Version 6 Release 3	xv
GC24-6247-00, z/VM Version 6 Release 2	xv
GC24-6232-00, z/VM Version 6 Release 1	xv
Chapter 1. Servicing your system.	1
Overview of the service procedure	2
Merging service	3
Receiving service	3
Applying service	3
Reworking local service (If applicable).	3
Building new levels	3
Testing serviced components	3
Placing the serviced components into production	3
Component dependencies and service order	4
Out-of-Component requisites	4
Product Parameter File	4
Disks used to service z/VM	5
Chapter 2. Installing preventive or corrective service	7
Requirements and considerations	7
Ordering service	8
Steps to receive, apply, and build preventive or corrective service.	8
Steps for placing service into production	12
Chapter 3. Applying local service and local modifications.	15
Overview of the local modification process	16
Applying or creating a local modification	17
Reworking a local modification.	18
Applying local service.	19
Chapter 4. Running stand-alone builds	23
Rebuilding the CP, CMS or GCS nucleus	23
Rebuilding segments	23
Rebuilding the CMS saved system.	23
Chapter 5. Removing service	25
Removing a single PTF	25
Removing a local modification	25
Restoring removed service	26

Removing service levels	26
Chapter 6. Setting up VMSES/E licensed products to use the SERVICE command	29
Appendix A. Receiving service for non-VMSES envelopes	33
Appendix B. Testing service for the base components of z/VM	35
Testing components with new service	35
IPLing with a test level of the CPLOAD MODULE	36
Testing CMS with LE, the Callable Services Library and the Shared File System	37
Testing TCP/IP for z/VM Service	39
Additional testing notes	40
Appendix C. Testing service for preinstalled components and features	41
RACF Security Server	41
Directory Maintenance Facility (DirMaint)	41
Remote Spooling Communications Subsystem (RSCS)	41
Appendix D. Local modification example	43
Appendix E. Typical local modifications	45
Modifying the CMSINST segment	45
Updating the CP load list	47
Steps for updating the CP load list	48
Notices	51
Privacy policy considerations	53
Trademarks	53
Glossary	55
Bibliography	57
Where to get z/VM information	57
z/VM base library	57
z/VM facilities and features	58
Prerequisite products	59
Index	61

Figures

1. Disks used when installing service	2
---	---

Tables

1. Components, Features, and Products Supported by SERVICE and PUT2PROD 7
2. Test build disks and production disks. 35

About this document

This document describes how to service version 6 release 3 of IBM® z/VM® and includes the preventative service procedure, which utilizes the recommended service upgrade (RSU), and the corrective (COR) service procedure for all components. Local service and modifications, reworking local service, and supplemental information and procedures are also included.

Intended audience

This information is intended for individuals responsible for maintaining and servicing a z/VM system. It is assumed you understand the concepts of z/VM service and are familiar with the syntax and usage of the z/VM service tools. If you do not understand these concepts or service tools, see the *z/VM: VMSES/E Introduction and Reference*.

Conventions and terminology

In this book, the term "MAINT $_{vrm}$," where vrm is the version, release, and modification level of the z/VM system, refers to the user ID that you must use to service all z/VM and preinstalled products, features, and components.

Various conventions are used to depict what you should type and what system responses you might see. Procedures will use the following conventions:

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

attach <i>tapeaddr</i> * 181	<i>tapeaddr</i>
TAPE <i>tapeaddr</i> ATTACHED TO <i>userID</i> 181	is the address of the tape drive(s)
Ready; T=	where the z/VM System DDR tapes
Ready; T=	will be mounted.
<i>n.nn/n.nn hh:mm:ss</i>	<i>userID</i>
	is the first-level user ID logged on to
	in the previous substep.

- 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
```

- Bold font indicates exactly what you should type.

Example: The following shows a command you would type:

disconnect

- 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 Enter:

ENTER

- A vertical bar (|) indicates you will receive or enter one of the values within the braces ({}).

Example: The following indicates sample output where you might receive one of two responses:

```
{MDREST|ECKDREST}: WROTE nnnn {BLOCKS|TRACKS} ON addr, RC=0
```

In this example, you would actually receive *one* of the following two responses:

```
MDREST: WROTE nnnn BLOCKS ON addr, RC=0  
ECKDREST: WROTE nnnn TRACKS ON addr, RC=0
```

Using the online HELP facility

You can receive online information about the service tools provided by VMSES/E 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
```

You can also display information about a message by entering one of the following commands:

```
help msgid or help msg msgid
```

For example, to display information about message VMF2130W, you can enter one of the following commands:

```
help vmf2130w or help msg vmf2130w
```

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
```

Where to find more information

For information about related documents, see “Bibliography” on page 57.

Links to other documents and web sites

The PDF version of this document contains links to other documents and web sites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a web site works only if you have access to the Internet. A document link is to a specific edition. If a

new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Go to IBM z/VM Reader's Comments (www.ibm.com/systems/z/os/zvm/zvmforms/webqs.html).
3. Mail the comments to the following address:
IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.
4. Fax the comments to us as follows:
From the United States and Canada: 1+845+432-9405
From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
z/VM V6.3 Service Guide
GC24-6247-01
- The topic name or page number related to your comment
- The text of your comment

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 use the personal information that you supply only to contact you about the issues that you submit to IBM.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See IBM: z/VM Service Resources (www.ibm.com/vm/service/).
- Go to IBM Support Portal (www.ibm.com/support/entry/portal/Overview/).

Summary of Changes

This document contains terminology, maintenance, and editorial changes. Technical changes are indicated by a vertical line to the left of the change. Some z/VM product changes might be provided through service by program temporary fixes (PTFs) for authorized program analysis reports (APARs), which also might be available for some prior releases.

GC24-6247-01, z/VM Version 6 Release 3

This edition supports the general availability of z/VM V6.3.

GC24-6247-00, z/VM Version 6 Release 2

This edition supports the general availability of z/VM V6.2.

This book has been reorganized for this release, and some chapters and sections have been deleted.

In addition to the reorganization, new and changed information includes the following:

- The new *MAINTorm* user ID is described in “Requirements and considerations” on page 7.

GC24-6232-00, z/VM Version 6 Release 1

This edition supports the general availability of z/VM V6.1.

Chapter 1. Servicing your system

This book contains procedures you can use to install the following kinds of service to z/VM and supported products that are in VMSES/E format:

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

VMSES/E provides service in Corrective (COR) service format or preventive service format in the form of the Recommended Service Upgrade (RSU). Corrective service (COR) contains requested PTFs and their requisite PTFs, if so specified at order time. The Recommended Service Upgrade (RSU) contains pre-applied service (PTFs) along with their associated prebuilt objects.

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 on a COR or RSU. When it is absolutely necessary to apply service before it is available in PTF form, or when you need a local modification to tailor your system environment, you must apply the service locally.

Service for IBM Licensed Products that are in VMSES/E format can also be installed using the automated service commands SERVICE and PUT2PROD.

In a Single System Image (SSI) cluster, the disks used for servicing a single release of z/VM are shared by all member systems running that release. Therefore the SERVICE command has to be run only once for a given level of z/VM. Because each member system has its own set of production disks, the PUT2PROD command must be run on each member system that is running the level of z/VM that you serviced.

Note:

1. Preventative and COR service is available on tape or electronically as service envelopes (with a file type of SERVLINK).
For more information about ordering and receiving service, see IBM: z/VM Service Resources (www.ibm.com/vm/service/).
2. *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.
3. 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 service procedure

When you service a z/VM component, the following main service tasks must be completed:

- Merging service
- Receiving service for the component
- Applying service to the component
- Reworking local service (if applicable)
- Building the new levels for the component
- Testing the new service
- Placing serviced components into production.

There are general commands that automate these tasks. The SERVICE command will receive, apply, identify and build the new service. If rework has been identified, the LOCALMOD command should be used to accomplish this. Finally, after you have tested your new service, the PUT2PROD command is used to place the service into production.

The SERVICE, LOCALMOD and PUT2PROD commands are referred to as the automated service commands. Throughout the service process, the automated service commands are used to manage and update your software inventory files.

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

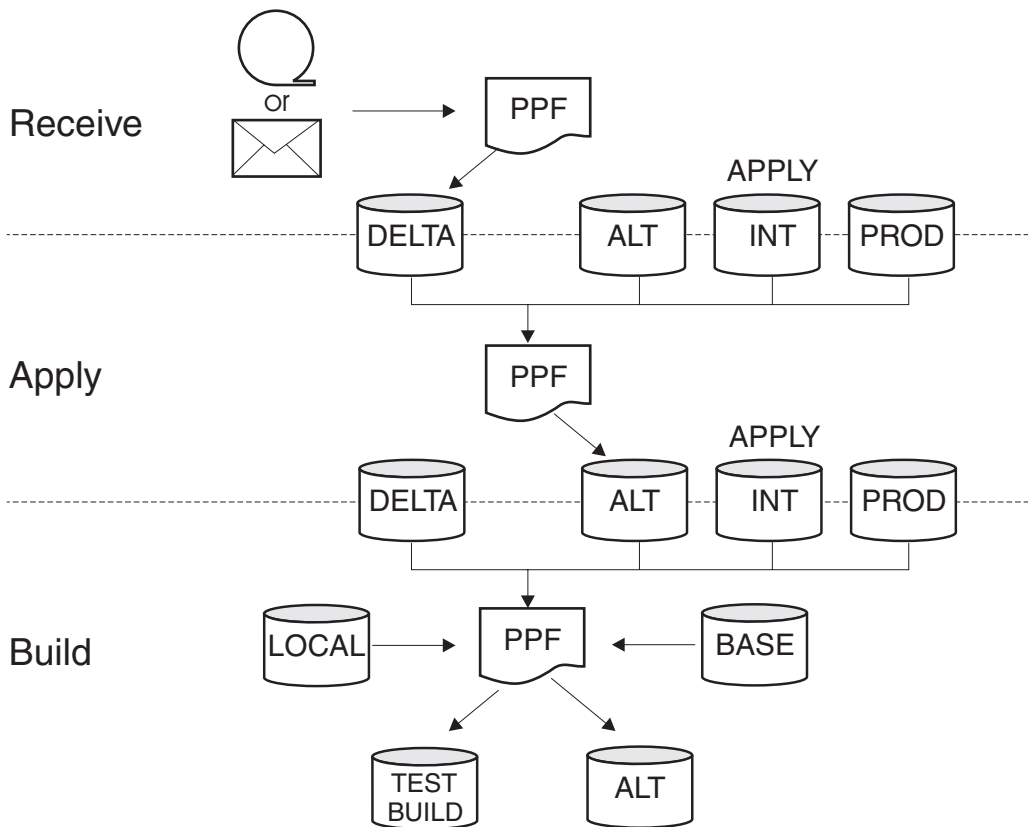


Figure 1. Disks used when installing service

The service and inventory disks are described in the detail in *z/VM: VMSES/E Introduction and Reference*.

Merging service

The VMFMRDSK command clears 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.

Receiving service

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

Applying service

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

Reworking local service (If applicable)

The LOCALMOD command is used to manage and rework local service. All local service must be entered into the software inventory to allow VMSES/E to track the changes and build them into the system. **If a local change is not added to the software inventory, VMSES/E will not build it into the system.**

It is possible that IBM service will affect your current local service and that you will need to rework your local service.

Building new levels

The VMFBLD command generates a serviced level of an object on a test build disk. The serviced objects can be tested before placing them into production.

Testing serviced components

When service is installed for a component, the serviced parts are built and put onto test disks. This gives you the opportunity to test the serviced parts before you put them into production. For information on testing service for the preinstalled features of z/VM (such as TCP/IP and DirMaint), see Appendix C, "Testing service for preinstalled components and features," on page 41. For the base components of z/VM (CMS, CP, and others), Appendix B, "Testing service for the base components of z/VM," on page 35 contains information on processes you can adapt for testing each component. The appendices also contain tables defining the relationships between test and production disks for each base component and for preinstalled features.

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), such as CMS and GCS.
- Place the CP nucleus, CMS nucleus and GCS nucleus into production.
- Copy the serviced objects to their production disks.

- Merge service inventory to the production apply disks.

Component dependencies and service order

Some components of z/VM use functions provided by other z/VM components for normal operations. Consequently, components can have dependencies on one another. These dependencies create a service order for the z/VM base components.

IBM recommends that you service z/VM base components in the following order. The SERVICE command respects this order when it services multiple components at one time.

1. VMSES/E: 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: 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. REXX/VM is part of CMS and GCS nuclei, so it must be serviced before CMS and GCS are serviced.
3. Language Environment
4. CMS: 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. When building CMS, it is important to note that the VM callable services libraries, shared file system (SFS) modules, REXX/VM, and CMS nucleus are all functionally dependent on each other.
5. CP
6. GCS: 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: These components can be serviced in any order.

The other preinstalled components of z/VM do not have a rigid service order. If you are applying service to multiple components at one time, such as installing RSU service, the SERVICE command will install the service for all the z/VM base components first and then will install the other components according to the order that they are found in the service media.

Out-of-Component requisites

The service stream may introduce other dependencies between components not shown in “Component dependencies and service order.” When a PTF requires service to several components, it introduces one or more out-of-component requisites that are identified when SERVICE calls VMFAPPLY. These out-of-component requisites are listed in the SERVICE and VMFAPPLY message logs.

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

Product Parameter File

The Product Parameter File (PPF) controls how service is installed for each component. Each component has a separate source product parameter file (\$PPF) that contains the component’s service media layout, associated user IDs, disks needed for service, and component build information.

In addition to the base \$PPF for each component, an override \$PPF file, SERVP2P \$PPF, is provided that identifies the component names of those components that are preinstalled on the z/VM system deliverable.

Disks used to service z/VM

The following section describes how the service disks are used during the installation of service.

Each component uses a unique set of disks or SFS directories for service and local modifications. These disks are defined for each component in the MDA section of the PPF for that component.

The following disks are the types of service disks that you can use:

TASK

The task disks contain any files that you want accessed before the service disks defined for a component.

LOCALMOD

The local disk (the default is *nC4*) contains any local modifications or local service you have installed.

LOCALSAM

The local sample disk (the default is *nC2*) contains the sample and example files you receive from IBM, such as the LOGO CONFIG and DMSNGP ASSEMBLE files.

DELTA

The delta disk (the default is *nD2*) contains all the files received from your service media, 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).

APPLY

The files on 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 multiple default disks defined as apply disks for each component in z/VM: the alternate apply disk (the default is *nA6*), the intermediate apply disk (the optional *nA4* disk), and the production apply disk (the default is *nA2*).

BUILD_n

The build disks contain all the usable forms (objects) built by VMFBLD. Each component has at least one test build disk and one production build disk.

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 other disks that need to be accessed after the service disks in order to service and build a component.

Servicing Your System

Chapter 2. Installing preventive or corrective service

This topic describes how to install preventive (RSU) or corrective (COR) service. You will use the `SERVICE` command to determine your current RSU service level, and then to receive, apply, and build the RSU or COR service that you ordered. Once you have installed and tested the service, you will use the `PUT2PROD` command to place the service into production. In an SSI cluster, you will need to run the `PUT2PROD` command on each system that is a member of the cluster and is running the z/VM base component or feature that was serviced.

The `SERVICE` and `PUT2PROD` commands support servicing the z/VM preinstalled components and features, and any other products that are in VMSES/E format *and* that have been updated to use the new functions of the `PUT2PROD` command. The z/VM preinstalled components and features are listed in Table 1.

Table 1. Components, Features, and Products Supported by SERVICE and PUT2PROD

Product	Component name
VMSES/E	VMSES
REXX™/VM	REXX
Language Environment	LE
CMS	CMS
CP	CP
GCS	GCS
Dump Viewing Facility	DV
TSAF	TSAF
AVS	AVS
RSCS Networking for z/VM	RSCS
TCP/IP	TCPIP
OSA/SF for VM	OSA
Directory Maintenance Facility	DIRM
RACF Security Server for z/VM	RACF
Performance Toolkit for VM	PERFTK
HCD and HCM for z/VM	VMHCD

Requirements and considerations

Before you start the service process, note the following requirements and considerations:

- The `MAINTorm` user ID, where *orm* is the version, release, and modification level of the z/VM system, must now be used rather than the `MAINT` user ID, to service all z/VM and preinstalled products, features, and components. You can use `MAINTorm` to run the `SERVICE` command from any system in a Single System Image (SSI) cluster. Placing the service into production must be done on each system in a SSI cluster. In an SSI cluster with multiple levels of z/VM, you should log on to `MAINTorm` to run the automated service commands (`SERVICE`, `LOCALMOD`, etc.) on a member that is running z/VM *orm*.

Installing preventive (RSU) or corrective (COR) service

- You must use the same minidisk address for your A-disk when running the SERVICE and PUT2PROD commands. VMSES/E uses the A-disk as a work disk.
- If the DIRMAINT, OSASF, OSAMAIN, and TCPMAINT user IDs are logged on when you run PUT2PROD, these IDs will be logged off because PUT2PROD must access in read/write mode disks owned by these IDs.
- For more information about the use of the SERVICE and PUT2PROD commands, see *z/VM: VMSES/E Introduction and Reference*.
- For more information on using the SERVICE command to service VMSES/E-formatted program products that are not preinstalled as part of z/VM installation, see Chapter 6, “Setting up VMSES/E licensed products to use the SERVICE command,” on page 29.

Ordering service

Service is ordered through the IBM ShopzSeries productivity tool on the web, or by contacting the IBM Software Support Center for your country.

When you order service, you need to ensure that you get all the necessary pre-requisite service. To do this, specify your current RSU service level on your service order.

Use the SERVICE command with the STATUS operand to determine the current RSU service level for a specific component or product, or for all components and products in your VMSES/E software inventory. The SERVICE command queries your inventory files and reports back the current service levels.

Before you begin: You must have the software inventory disk accessed. By default, the software inventory disk is the 51D disk and it is accessed by the MAINT_{orm} user ID as filemode D.

To determine the RSU level of a component, issue the following command:

```
service compname status
```

where *compname* is component listed in your software inventory table.

Example: In this example, “1101” represents the RSU level that you would use when ordering service for CP.

```
service cp status ENTER
VMFSRV2195I SERVICE CP STATUS
VFSSRV2760I SERVICE processing started
VMFSRV1225I CP (6VMCPR20%CPSFS) status:
VMFSRV1225I Service Level RSU-1101
VMFSRV1225I Production Level RSU-1101
VMFSRV2760I SERVICE processing completed successfully
```

For more information on the SERVICE command, see *z/VM: VMSES/E Introduction and Reference*.

Steps to receive, apply, and build preventive or corrective service

In these steps, you will use the SERVICE command to receive, apply, and build the preventative (RSU) or corrective (COR) service.

Note:

Steps to receive, apply, and build preventive or corrective service

1. To service z/VM *vr*m, log on to MAINT*vr*m. In an SSI cluster, log on to MAINT*vr*m on any member system that is running z/VM *vr*m and run the SERVICE command. This has to be done only once per z/VM level.
2. The SERVICE command processes only one RSU or COR per invocation. It does, however, process all volumes of a multiple-volume RSU or COR in a single invocation.
3. If you received any additional preventative service planning (PSP) COR service with your RSU, install the RSU service before you install the PSP COR.

Perform these steps to receive, apply, and build the RSU or COR service:

1. Create a complete backup of your system. For more information on storing a backup copy of your z/VM system on tape or DASD, see *z/VM: Installation Guide*.

2. Log on as MAINT*vr*m.

```
logon maintvr  
:  
z/VM Vv.r.m   yyyy-mm-dd hh:mm
```

ENTER

```
:  
Ready; T=n.nn/n.nn hh:mm:ss
```

3. If the RSU or COR is on tape, continue with this substep. Otherwise, skip to step 4 on page 10.
 - a. Attach a tape drive (*tapeaddr*) to MAINT*vr*m at virtual device address 181.

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

- b. Mount volume 1 on your 181 tape drive. If you are using an automated tape library (ATL), see the automated tape library manuals.
If the RSU or COR has multiple volumes, perform one of the following actions:
 - Stack the RSU or COR volumes on the tape drive attached at virtual address 181.
 - Attach additional tape drives as 182, 183, etc., and mount each volume.
- c. IPL CMS.

```
ipl cms  
z/VM Vv.r.m   yyyy-mm-dd hh:mm
```

ENTER

```
Ready; T=n.nn/n.nn hh:mm:ss
```

- d. Issue the SERVICE command.
 - If all of the volumes are stacked on the tape drive attached at virtual address 181, issue the SERVICE command with no parameters:

```
service
```

- If each of the volumes are mounted on a separate tape drive, issue the SERVICE command with the following parameters:

Steps to receive, apply, and build preventive or corrective service

```
service all tapeaddr1 tapeaddr2 ... List the addresses of each tape volume.
                                     tapeaddr1
                                     Tape address for tape volume 1
                                     tapeaddr2 ...
                                     Tape address for tape volume 2, and
                                     so on.
```

You will see the following after issuing the SERVICE command:

```
VMFSRV2760I SERVICE processing started
:
VMFSRV2760I SERVICE processing completed successfully
Ready; T=n.nn/n.nn hh:mm:ss
```

Continue with substep 6 on page 11.

4. If the RSU or COR service is electronic or on CD-ROM, continue with this substep.
 - a. Retrieve the entire RSU or COR service. (The MAINT \overline{vrm} 500 minidisk can be used to store the service files.)
 - If your media is electronic, follow the instructions that were sent to you electronically.
 - If your media is CD-ROM, follow the instructions in the README file on the CD-ROM.

Note: You must preserve the file attribute of FIXED BINARY 1024 through all intermediate transports.

- b. IPL CMS.

```
ipl cms
z/VM Vv.r.m yyyy-mm-dd hh:mm
```

ENTER

```
Ready; T=n.nn/n.nn hh:mm:ss
```

- c. Access the minidisk containing the service files and decompress the VMSES/E envelope file.

```
access diskaddr c
Ready;
```

```
deterse envfn envft envfm = servlink =
```

You need to enter the DETERSE command for every envelope file you received with your order.

- d. Run SERVICE with the COR documentation envelope file to extract the COR service memo, and then use the VMFUPDAT command to display the memo. This step is for COR service only. The RSU documentation envelope file is a readable text file.

```
service all docenvfn
```

```
docenvfn
```

File name of the documentation envelope.

```
vmfupdat sysmemo
```

Displays the memos that were in the documentation envelope

Steps to receive, apply, and build preventive or corrective service

- e. Run SERVICE with the COR or RSU envelope.

```

service all envfn1 envfn2 ...
VMFSRV2760I SERVICE processing
started
:
:
VMFSRV2760I SERVICE processing
completed successfully
Ready; T=n.nn/n.nn hh:mm:ss
    
```

List the file names of each COR or RSU envelope. You can specify more than one envelope file only if the COR or RSU is multivolume.

envfn1

File name for the first envelope.

envfn2 ...

File name for the second envelope, and so on.

5. Enter VMFVIEW SERVICE to view the SERVICE message log.
6. If you received a nonzero return code from SERVICE, use the following table to help you determine the appropriate actions to take. Base your actions on the following table:

If you received . . .	Then perform the following . . .
Return code 4	<ol style="list-style-type: none"> 1. You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received. 2. Go to substep 7 on page 12.
Return code 6 and the message: VMFSUII2760I VMFSUFIN PROCESSING INCOMPLETE DUE TO LOCAL MODIFCATIONS	<ol style="list-style-type: none"> 1. You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received. 2. Use the VMFUPDAT SYSLMOD command to see which local modifications you need to rework. Rework the local modifications that were affected by service according to the information in Chapter 3, "Applying local service and local modifications," on page 15. For more information, see <i>z/VM: VMSES/E Introduction and Reference</i>. 3. After you complete the rework, use the VMFUPDAT SYSLMOD command to flag the local modification as reworked. 4. Restart by issuing the SERVICE command as displayed in message VMFSRV2310W. 5. Continue from substep 5.
A return code greater than 6	<ol style="list-style-type: none"> 1. You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received. 2. Correct all errors reported in the error messages. 3. Restart by issuing the SERVICE command as displayed in the message VMFSRV2310W. 4. Continue from substep 5.

Note:

- a. You can ignore message VMFSET2206W for minidisk 5E6.
- b. You can ignore the following messages and the associated VMF1966W message:
 - DMSLI0201W The following names are undefined: ISPLINK ARIPRDI
 - DMSLI0201W The following names are undefined: DMSDSCSC
 - DMSLI0201W The following names are undefined: DMSUSRX1 DMSUSRX2

Steps to receive, apply, and build preventive or corrective service

- DMSLI0202W Duplicate identifier messages associated with object IOACMAIN MODULE.
 - DMSLKD004W Warning messages issued messages associated with objects ILBONBL, ILBONTR, ILBOREC, ILBORNT, ILBOSND, ILBOSNT, and ILBOSSN.
 - DMSLI0994W Restrictive RMODE encountered in CSECT *filename*
7. Use the VMFUPDAT SYSMEMO command to review any memos that were received with the service.
 8. Test the service you just installed. If the service is for a base component of z/VM, see Appendix B, “Testing service for the base components of z/VM,” on page 35 for information on testing new service if you have not done so already. For preinstalled products, see Appendix C, “Testing service for preinstalled components and features,” on page 41. For all other products, refer to their respective program directories.

Steps for placing service into production

In these steps, you will run PUT2PROD to place the products that were processed by SERVICE into production. Do these steps only after you have tested, and are satisfied with, the service.

Note:

1. PUT2PROD will affect your production environment. It is recommended that all users be logged off. If you are placing CP service into production, the place into production steps should be performed as part of a planned system outage. The following list of production environment virtual machines will be logged off to write to minidisks or SFS directories they own.
 - DIRMAINT
 - OSASF
 - OSAMAINT
 - TCPMAINT
2. If you are running RACF Security Server and you receive the message VMFP2P1219E while running PUT2PROD, follow the instructions in “RACF Security Server” on page 41.
3. To put serviced code for z/VM *vrn* into production, log on to MAINT*vrn*. In an SSI cluster where more than one member is running the same level of software, log on to MAINT*vrn* on each member that is running z/VM *vrn* and run PUT2PROD. You may want to relocate any guests that can be relocated to other members of the SSI cluster before you run PUT2PROD on any specific member.
 1. Log on as MAINT*vrn*.

```
ENTER
logon maintvrn
:
:
z/VM Vv.r.m      yyyy-mm-dd hh:mm

ENTER
:
:
Ready; T=n.nn/n.nn hh:mm:ss
```

2. IPL CMS.

Steps for placing service into production

```
ipl cms  
z/VM Vv.r.m      yyyy-mm-dd hh:mm
```

ENTER

```
Ready; T=n.nn/n.nn hh:mm:ss
```

3. Run PUT2PROD.

```
put2prod  
VMFP2P2760I PUT2PROD processing started  
:  
VMFP2P2760I PUT2PROD processing completed successfully  
Ready; T=n.nn/n.nn hh:mm:ss
```

4. Enter VMFVIEW PUT2PROD to view the PUT2PROD message log.
5. If you received a nonzero return code from PUT2PROD, use the following table to help you determine the appropriate actions to take:

If you received . . .	Then . . .
Return code 4	<ol style="list-style-type: none">1. You can ignore any warning messages listed in the note below. Take appropriate actions for any other warning messages you receive.2. Go to substep 6.
A return code greater than 4	<ol style="list-style-type: none">1. You can ignore any warning messages listed in the note below. Take appropriate actions for any other warning messages you receive.2. If you received message VMFP2P1219E see "RACF Security Server" on page 41.3. Correct all errors reported in the error messages.4. Issue IPL CMS.5. Issue PUT2PROD.6. If you receive a nonzero return code, repeat substep 5.7. Go to substep 6.

Note: The following messages can be ignored:

- DMSDCS1083E Saved segment \$\$DMY\$\$ does not exist
 - DMSWLG292W Text data will be loaded at '20000'x in user area; user data may be overwritten.
6. If you serviced CP, shut down and re-IPL the system. If CP was not serviced, recycle servers as necessary.

```
shutdown reipl
```

ENTER

You have now completed the service procedure for one system. In an SSI cluster, repeat the PUT2PROD procedure for each additional member that is running the same level of software.

Chapter 3. Applying local service and local modifications

There are two types of local service: 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 delivery media (tape, electronic, or CD-ROM). When a severe problem arises and you cannot wait for a COR or RSU, you can get emergency service from IBM. This service can be sent to you on a tape, electronically, or read to you over the phone. In any case, the service is not in the format required by the SERVICE command. 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 you make to tailor your z/VM system. These updates can be made by you to customize configuration files, or they 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. Note that the disk used to store local modifications and local service is a shared service disk in an SSI cluster, but it is only shared by the members of the SSI cluster that are running the same level of z/VM. If you are running multiple levels of z/VM in your SSI cluster, you might need different local modifications for the different levels of z/VM.

There are three ways that parts can be modified. Source-maintained parts are modified by changing the source, with an update file, an AUX file, and a CNTRL file. Replacement-maintained parts are modified by replacing the part with an updated version. Finally, when necessary, some parts (such as text files) can be modified by directly changing the object code they contain.

The step-by-step procedures for applying local service and local modifications to your z/VM system, outside the normal service process, are described here.

This section also contains the step-by-step procedures for reworking changes installed through the local modification process that might have been affected by new IBM service.

Reworking local modifications might be necessary if the service you receive from IBM affects 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 modification and then, if the modification is still needed, rework it and rebuild the affected parts.

Examples for applying local modifications to the CP load list and CMSINST can be found in the appendixes. If you want to update the CP load list see Appendix E, “Typical local modifications,” on page 45 for instructions on using the GENCPBLS command. If you plan to change which EXECs are placed in the CMS installation segment see Appendix D, “Local modification example,” on page 43.

Overview of the local modification process

The following is an overview of the local modification process. The LOCALMOD command automates most of these steps.

1. Log on to the MAINT_{vr}m user ID on a system that is running z/VM *vr*m.
2. Receive Local Service or Modifications.
If you have IBM local service, load the service to the MAINT_{vr}m A-disk (191)
3. Apply Local Service to Source-Maintained parts using the LOCALMOD command.
 - 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 using the LOCALMOD command.
 - 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.
5. Rebuild objects using the SERVICE command. This command will:
 - 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.
6. Place the changes into production using the PUT2PROD command.

Guidelines for using update files

The following are guidelines for using update files for source updates and patches.

1. Keep each fix in a separate update file. You can have several update files containing one fix, but you must never have several fixes in one update file. Each fix should have a unique identifier for control purposes. This identifier is the file type of the update file. If the same fix applies to several parts, there should be a update file for each part, all with the same file type.
2. Keep all local fix descriptions for the same part in the same AUX file, unless a fix applies to a different control file level.

You should not distribute local fixes for the same part over AUX files (different control file levels) arbitrarily. Local service can be distributed over several control files to maintain different service levels with a single AUX and update file structure. You can build each level from a different control file that contains only the desired level identifiers.
3. Never place local fixes in AUX files from IBM. In other words, keep your local service separate from IBM service. Local service should be easily distinguished from IBM service and should always be applied last.
4. Patches to text files should be applied only when no source file is available. If you do apply text file patches when source code is available, you should convert them to source updates and reassemble them before you move them from a test environment to your production system.

Building local source updates on top of local text file patches for the same part will lead to confusion.
5. Do not place the names of update files in the main control file. Place update file names in AUX files.

Applying or creating a local modification

Use the LOCALMOD command to create and apply a local modification to any component or product defined in the System-Level Service Update Facility (VM SYSSUF) software inventory table.

Note: In an SSI cluster, you have to run the LOCALMOD and SERVICE commands on only one member system. However, on each member system where you want to put your local modification in to production you must log on with MAINT_{orm} and then run the PUT2PROD command. If you are running multiple levels of z/VM in your SSI cluster, and you will be putting your local modification on members that are running different levels of z/VM, you must run the local modification commands for each release of z/VM. For example, if you have an SSI cluster with one member running z/VM 6.2.0 and one member running z/VM 6.3.0, you must log on to MAINT620 on the system running z/VM 6.2.0 and run the local modification commands; then you must log on to MAINT630 on the system running z/VM 6.3.0 and run the local modification commands there. Once this is done you must put each member into production in the usual manner.

Before you begin: You must have the software inventory disk accessed. By default, the software inventory disk is the 51D disk and it is accessed as D.

1. Log on to the MAINT_{orm} user ID.
2. Create and apply the local modification.
 - a. Run LOCALMOD:

Apply or create a local modification

`localmod compname partfn partft`

compname

Component name associated with the product you are modifying (for example one of the COMPNAME values in Chapter 2, "Installing preventive or corrective service," on page 7).

partfn

File name of the source part that is to be modified.

partft

File type of the source part that is to be modified.

- b. Reply to any prompt messages.
- c. Make your changes to the displayed file.
- d. File your changes.

====> **file**

Enter **file** on the XEDIT command line.

- e. Repeat this entire step for any additional local modifications that you need to make.
3. Run SERVICE, for each component processed, to build the local modification(s).

`service compname build`

4. Run PUT2PROD to put the local modification(s) into production.

`put2prod`

Note: You can use PUT2PROD only for products that have been updated to use the new PUT2PROD functions. For other products, follow that product's instructions (found in the product's program directory) to place its service into production.

Reworking a local modification

Use the LOCALMOD command to rework a local modification to any component or product defined in the System-Level Service Update Facility (VM SYSSUF) software inventory table.

Note: In an SSI cluster, you have to run the LOCALMOD and SERVICE commands on only one member system. However, on each member system where you want to put your local modification in to production you must log on with MAINT_{orm} and then run the PUT2PROD command. If you are running multiple levels of z/VM in your SSI cluster, and you will be putting your local modification on members that are running different levels of z/VM, you must run the local modification commands for each release of z/VM. For example, if you have an SSI cluster with one member running z/VM 6.2.0 and one member running z/VM 6.3.0, you must log on to MAINT620 on the system running z/VM 6.2.0 and run the local modification commands; then you must log on to MAINT630 on the system running z/VM 6.3.0 and run the local modification commands there. Once this is done you must put each member into production in the usual manner.

Before you begin: You must have the software inventory disk accessed. By default, the software inventory disk is the 51D disk and it is accessed as D.

1. Log on to the MAINT_{orm} user ID.
2. Rework the local modification.
 - a. Run LOCALMOD:

```
localmod compname partfn partft (rework
```

compname

Component name associated with the product you are reworking (for example one of the COMPNAME values in Chapter 2, "Installing preventive or corrective service," on page 7).

partfn

File name of the source part that is to be reworked.

partft

File type of the source part that is to be reworked.

- b. Reply to any prompt messages.
- c. Make your changes to the displayed file.
- d. File your changes.

```
====> file
```

Enter **file** on the XEDIT command line.

- e. Repeat this entire step for any additional local modifications that you need to rework.
3. Run SERVICE, for each component processed, to build the local modification(s). If you received a return code 6 (rework local modifications) from a SERVICE invocation, enter the following:

```
service restart
```

Otherwise, enter the following:

```
service compname build
```

4. Run PUT2PROD to put the local modification(s) into production.

```
put2prod
```

Note: You can use PUT2PROD only for products that have been updated to use the new PUT2PROD functions. For other products, follow that product's instructions (found in the product's program directory) to place its service into production.

Applying local service

Use the LOCALMOD command to apply local service to any component or product defined in the System-Level Service Update Facility (VM SYSSUF) software inventory table. Local service is defined as any service received from IBM that is not on a COR or RSU tape (or envelope).

Apply Local Service

Note: In an SSI cluster, you have to run the LOCALMOD and SERVICE commands on only one member system. However, on each member system where you want to put your local modification in to production you must log on with MAINT_{orm} and then run the PUT2PROD command. If you are running multiple levels of z/VM in your SSI cluster, and you will be putting your local modification on members that are running different levels of z/VM, you must run the local modification commands for each release of z/VM. For example, if you have an SSI cluster with one member running z/VM 6.2.0 and one member running z/VM 6.3.0, you must log on to MAINT620 on the system running z/VM 6.2.0 and run the local modification commands; then you must log on to MAINT630 on the system running z/VM 6.3.0 and run the local modification commands there. Once this is done you must put each member into production in the usual manner.

Before you begin: You must have the software inventory disk accessed. By default, the software inventory disk is the 51D disk and it is accessed as D.

1. Log on to the MAINT_{orm} user ID.
2. Apply the local service.
 - a. Run LOCALMOD:

```
localmod compname partfn partft servfn servft
```

compname

Component name associated with the product you are modifying (for example one of the COMPNAME values in Table 1 on page 7).

partfn

File name of the part to be serviced.

partft

File type of the part to be serviced.

If the service is an update, *partft* is the file type of the service part, for example ASSEMBLE. If the service is a replacement part, *partft* is the file type of the part to be replaced, for example TEXT.

servfn

File name of the service received from IBM.

servft

File type of the service received from IBM.

- b. Reply to any prompt messages.
 - c. Repeat this entire step for any additional local service.
3. Run SERVICE, for each component processed, to build the local service.

```
service compname build
```

4. Run PUT2PROD to put the local service into production.

```
put2prod
```

Note: You can use PUT2PROD only for products that have been updated to use the new PUT2PROD functions. For other products, follow that product's instructions (found in the product's program directory) to place its service into production.

Apply Local Service

Chapter 4. Running stand-alone builds

Occasionally you might need to rebuild part of your system, even if you haven't installed any new service. This section documents how to perform some of the more common build procedures.

Note: In an SSI cluster you have to run any SERVICE commands on only one member. However, PUT2PROD commands must be run on each member where a new build is needed.

Rebuilding the CP, CMS or GCS nucleus

Note: Rebuilding the CMS nucleus also rebuilds the ZCMS nucleus.

1. Log on to the MAINT`vr`m user ID on the system running z/VM `vr`m where you want to rebuild the nucleus.

2. Run the SERVICE command to build the nucleus:

```
service compname b1dnuc
```

where `compname` is CP, CMS or GCS.

3. Run the PUT2PROD command to make the new nucleus available:

```
put2prod
```

4. If you rebuilt the CP nucleus and you are running in an SSI cluster, relocate any guests that can be relocated to other members of the SSI cluster.

5. If you rebuilt the CP nucleus, shutdown and re-IPL the system:

```
shutdown reipl
```

6. If you rebuilt the CMS nucleus or the GCS nucleus, recycle any servers that need to use the new nucleus.

Rebuilding segments

1. Log on to the MAINT`vr`m user ID on the system running z/VM `vr`m where you want to rebuild the segments.

2. Run the PUT2PROD command. To rebuild all segments, issue:

```
put2prod segments all
```

Or, to rebuild a specific segment, issue:

```
put2prod segments segname
```

where `segname` is the name of the segment you want rebuild.

3. Recycle any servers that need to start using the new segments.

Rebuilding the CMS saved system

Note: Rebuilding the CMS saved system also rebuilds the ZCMS saved system.

1. Log on to the MAINT`vr`m user ID on the system running z/VM `vr`m where you want to resave CMS.

2. Run the PUT2PROD command to resave CMS:

```
put2prod savecms
```

Rebuilding Segments

3. Recycle any servers that need to start using the new saved system.

Chapter 5. Removing service

You can remove (back out) service levels or individual PTFs by un-applying the appropriate PTFs or groups of PTFs (service levels) 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. See *z/VM: VMSES/E Introduction and Reference* for a description of the VMFREM command.

The following sections identify the steps required to remove single PTFs, local modifications, and service levels.

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 servp2p compname
```

2. Remove the PTF and its dependents.

```
vmfrem ppf servp2p compname ptf ptfnum (remopts)
```

Where:

ptfnum is the PTF number of the PTF you are removing.

remopts are the remove options. You can specify UNAPPLY or UNRECEIVE or neither. UNAPPLY is the default.

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. Use the SERVICE command to rebuild the parts without the PTF that was removed:

```
service compname build
```

4. Put the rebuilt parts into production:

```
put2prod
```

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 servp2p compname
```

Removing Service

2. Remove the local modification.

```
vmfrem ppf servp2p compname mod modid (remopts
```

Where:

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

remopts are the remove options. You can specify UNAPPLY or UNRECEIVE or neither. UNAPPLY is the default.

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. Use the SERVICE command to rebuild the parts without the local modification:

```
service compname build
```

4. Put the rebuilt parts into production:

```
put2prod
```

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 removed by VMFREM was put in an 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.

2. Re-apply and build the service.

Note: If the UNRECEIVE option was specified on the VMFREM command when the service was removed, the service must be reinstalled using the normal service process. If this is the case, skip this step and see Chapter 2, "Installing preventive or corrective service," on page 7 for more information.

```
service compname norec
```

3. Put the restored service into production:

```
put2prod
```

Removing service levels

The following service levels are supported for base z/VM components:

- The alternate (current) service level
- The intermediate (previous) service level
- The oldest (production) service level.

Each service level is located on a separate apply disk. Most other products and features have only two apply disks and therefore support only two service levels.

APPLY (2A6)
Newest service level

APPLY (2A4)
Intermediate service level

APPLY (2A2)
Oldest service level

If you want to back out the newest service level and use only the previous service level, you must perform the following steps for one service level.

If you want to back out both the current and previous levels and use only the oldest 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 as well. 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 servp2p compname
```

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

```
vmfrem ppf servp2p 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
                                             previous levels)
```

If you get the following message during remove processing, you must decide how to proceed and then respond to message VMFREM2131R.

```
VMFREM2130W VMFREM has encountered the following
missing parts while attempting to remove service levels.
```

If you respond to continue, you must restore the missing parts before running VMFBLD. This can be done in one of the following ways:

- Retrieve them from a product, service, or backup tape.
 - Generate them, if source maintained, by using VMFHLASM or VMFEXUPD.
3. Use the SERVICE command to rebuild the parts without the service level that was removed:

```
service compname build
```

4. Put the rebuilt parts into production:
put2prod

Removing Service

Chapter 6. Setting up VMSES/E licensed products to use the SERVICE command

You can use the SERVICE command to install new service for licensed products that you have installed using VMSES/E. To use the SERVICE command for these products, you must update the System-Level Service Update Facility (VM SYSSUF) software inventory file by using the VMFUPDAT command.

The products that are preinstalled with the z/VM system are already included in the software inventory file. For a list of the preinstalled products, see Table 1 on page 7.

In an SSI cluster, if a licensed product is installed such that the service disks are shared between all members of the cluster, you will have to log on to only one member of the cluster and run the SERVICE command. You will still have to put serviced files for the licensed product into production on each member. If the service disks are not shared by all members, you will have to log on to the maintenance ID on each member to run SERVICE, and then put the new service into production.

Note: Some licensed products have also been updated to use new functions provided by the PUT2PROD command. Putting these products in the software inventory file enables the products to use the new PUT2PROD functions.

Before you begin:

- The licensed product that you are adding to the VM SYSUF table must be installed on the level of z/VM that owns the table you are updating. Refer to the documentation for the licensed product for installation instructions.
- You must have the software inventory disk accessed. By default, the software inventory disk is the 51D disk, and it is accessed as D.

Perform the following steps to set up the new licensed products:

1. Log on as MAINT \overline{vrm} .

```
ENTER
logon maint $\overline{vrm}$ 
:
Ready; T= $n.nn/n.nn$  hh:mm:ss
```

2. Invoke VMFUPDAT to update the VM SYSSUF file.

```
vmfupdat syssuf
```

Setting up VMSES/E licensed products to use SERVICE

```

*** Update SYSSUF Table Entries ***

Update any PPF/component name or YES|NO field. To change all occurrences
of a PPF name in the table replace both ***** fields with PPF names.

Compname          Prodid  Servlev  Prodlev  Description
-----
ADSM              5654A09A  RSU-0101  RSU-0101  ADSTAR Distributed Storage Manager for
:INSTALL  YES      :INSPPF  SERVP2P  ADSM
:BUILD    YES      :BLDPPF  SERVP2P  ADSM
:INCLUDE  YES      :P2PPPF  SERVP2P  ADSMP2P
AVS              6VMAVS30  000-0000  000-0000  AVS for z/VM 6.2.0z/VM 6.2.0
:INSTALL  YES      :INSPPF  SERVP2P  AVS
:BUILD    YES      :BLDPPF  SERVP2P  AVS
:INCLUDE  YES      :P2PPPF  SERVP2P  AVSP2P
CMS              6VMCMS30  000-0000  000-0000  CMS for z/VM 6.2.0z/VM 6.2.0
:INSTALL  YES      :INSPPF  SERVP2P  CMS
:BUILD    YES      :BLDPPF  SERVP2P  CMS
:INCLUDE  YES      :P2PPPF  SERVP2P  CMSP2P

Change PPF name ***** to *****

Page 1 of 9

PF1=HELP  PF3/PF12=Quit  PF5=Process  PF6=VMFSUFTB  PF8=Forward

```

- Press F6 to refresh the VM SYSSUF table from the information on your system inventory disk (51D). This will locate files that were created on the system inventory disk when you installed your new product(s), extract information from those files, and then add the information about the new product(s) to the VM SYSSUF table. the table will then be redisplayed.
- Locate the new products to be used with SERVICE by scrolling forward.

Example: In the following example, HLASM is the product that was added; PF8 was used to scroll through the file to the entry for HLASM.

```

*** Update SYSSUF Table Entries ***

Update any PPF/component name or YES|NO field. To change all occurrences
of a PPF name in the table replace both ***** fields with PPF names.

Compname          Prodid  Servlev  Prodlev  Description
-----
HLASM             5696234E  000-0000  000-0000  HIGH LEVEL ASSEMBLER FOR MVS & VM &
:INSTALL  YES      :INSPPF  5696234E  HLASM
:BUILD    YES      :BLDPPF  5696234E  HLASM
:INCLUDE  YES      :P2PPPF
ICKDSF           5684042J  011-0011  011-0011  ICKDSF DEVICE SUPPORT FACILITIES R17
:INSTALL  YES      :INSPPF  SERVP2P  ICKDSF
:BUILD    YES      :BLDPPF  SERVP2P  ICKDSF
:INCLUDE  YES      :P2PPPF  SERVP2P  ICKDSFP2P
LE               6VMLEN20  000-0000  000-0000  IBM Language Environment for zVM
:INSTALL  YES      :INSPPF  SERVP2P  LE
:BUILD    YES      :BLDPPF  SERVP2P  LE
:INCLUDE  YES      :P2PPPF  SERVP2P  LEP2P

Change PPF name ***** to *****

Page 4 of 10

PF1=HELP  PF3/PF12=Quit  PF5=Process  PF7=Backward  PF8=Forward

```

- If necessary, change the :INSPPF and :BLDPPF fields to the PPF name you are using to service this product. The :P2PPPF field will be blank. PUT2PROD will use the PPF and component information from the :BLDPPF field so it is not necessary to update the :P2PPPF field unless you are instructed to do so by the product's Program Directory.
- Press F5 to process. The VM SYSSUF table is updated and you exit VMFUPDAT.

Setting up VMSES/E licensed products to use SERVICE

You have completed the update procedure. You can now use SERVICE to apply service for new product(s) you added. For more information on using SERVICE, see “Steps to receive, apply, and build preventive or corrective service” on page 8.

|
|
|
|

Note: If you are running multiple levels of z/VM in your cluster, you must add your licensed product to the VM SYSSUF table for each release of z/VM. You should log on to each MAINT*orm* user ID in your cluster (for example, MAINT620 and MAINT630) and perform steps 1 thru 6 on each user ID.

Appendix A. Receiving service for non-VMSES envelopes

This appendix describes how to receive service for non-VMSES envelopes. The procedures in the main sections of this book cannot be used for applying non-VMSES service, so you should refer to the documentation that comes with the envelope and to this appendix.

1. Determine whether the service files are in VMSES or non-VMSES format. Base your action on the file name and file extension of the service files as they were delivered.

File name and file extension	Explanation
VLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOCS	These files are VMSES format service documents.
VPTF <i>n..n</i> .bin RPTF <i>n..n</i> .bin <i>ordernum</i> .SHIPTFSS <i>ordernum</i> .SHIPRSU <i>n</i>	These file are VMSES format service files.
RLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOC	These files are RSU service documents. Once uploaded, they are plain text files. No further action is needed.
NLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOCN	These files are non-VMSES format service documents.
NPTF <i>n..n</i> .bin <i>ordernum</i> .SHIPTFSN	These files are non-VMSES format service files.

2. 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).

3. Read the memo-to-users for each product that you plan to service.
4. Receive the service for non-VMSES PTF envelopes *nsptfenofn* SERVLINK.

Receiving Service For Non-VMSES Envelopes

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.

Appendix B. Testing service for the base components of z/VM

When service is installed for a component, the updated parts are built and stored on a test build disk. The test build disks for a component are defined in the PPF for that component, and contain all the files that are on a production disk, plus the new versions of the parts that were serviced. By putting the new parts onto a test build disk, the new parts can be tested before they are moved into production.

Testing components with new service

In general, new service is tested by logging on to a user ID and accessing the test build disks instead of the production disks, and then running some programs that will use serviced parts. The relationship between test build disks and production disks for the z/VM base components is described in Table 2. For other products, these relationships are defined in each product's program directory.

Table 2. Test build disks and production disks

Component	Disk owner (user ID)	Test build disks	Disk owner (user ID)	Production disks
VMSES/E	MAINT ν rm	490, 493, 49D, 5E6	MAINT	190, 193, 19D, 402, 5E5
REXX/VM	MAINT ν rm	490, 493	MAINT	190, 193
LE	6VMLEN20	49E, 4DD	MAINT	19E, 19D, 402
CMS	MAINT ν rm	490, 890, 493, 49D	MAINT	190, 990, 193, 19D, 402
	PMAINT	550	PMAINT	551
CP	MAINT ν rm	490, 493, 49D, CF2	MAINT	190, 193, 19D, 402, CF1
	PMAINT	550	PMAINT	551
GCS	MAINT ν rm	493, 49D	MAINT	193, 19D, 402
Dump Viewing Facility	MAINT ν rm	490, 493, 49D	MAINT	190, 193, 19D, 402
TSAF	MAINT ν rm	493, 49D	MAINT	193, 19D, 402
AVS	MAINT ν rm	493, 49D	MAINT	193, 19D, 402
TCPIP	ν VMTCPrm	491, 492	TCPMAINT	591, 592

Note: The 49D is the test help disk. Its contents are copied to both the 19D (Help) and 402 (Upper Case Help) disks.

To test a new CPLOAD MODULE, you will need to IPL your system with the CPLOAD MODULE that was built on your test parm disk (MAINT ν rm CF2). See "IPLing with a test level of the CPLOAD MODULE" on page 36 for more information.

Testing service for the base components of z/VMz/VM

To test a new CMS nucleus, you can issue the command IPL 490 instead of IPL 190 or IPL CMS.

IPLing with a test level of the CPLOAD MODULE

When you install service for CP with the SERVICE command, VMSES/E puts the serviced CPLOAD MODULE on the MAINT orm CF2 disk. To test the new CPLOAD MODULE you will need to IPL your system using that CPLOAD MODULE.

To IPL your test CPLOAD MODULE from the CF2 parm disk, you might need to change the DEVICE NUMBER on the Stand Alone Program Loader (SAPL) panel to the address of the volume where the CF2 parm disk is defined. By default the CF2 disk is defined on the DASD volume with label ormRL1 for systems installed on 3390 DASD, or with label ormRES for systems installed on SCSI volumes. For a system installed on 3390 DASD, this will not be the same volume where the CF1 and CF3 parm disks are located.

Note: If you are running the RACF Security Server feature of z/VM and it was also serviced, see "RACF Security Server" on page 41.

The following example shows the SAPL Panel as it is displayed when you IPL the production CPLOAD MODULE from the CF1 parm disk, with LOADPARAM consaddr :

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 6 RELEASE 2.0
DEVICE NUMBER:  A666      MINIDISK OFFSET:      39  EXTENT:  1
MODULE NAME:    CPLOAD    LOAD ORIGIN:      1000
-----IPL PARAMETERS-----
fn=SYSTEM ft=CONFIG pdnum=1 pdvo1=A667 cons= $\text{consaddr}$ 
-----COMMENTS-----
9= FILELIST 10= LOAD 11= TOGGLE EXTENT/OFFSET
```

To IPL with the test CPLOAD MODULE located on the CF2 parm disk, change the DEVICE NUMBER field to the device address where the CF2 parm disk is located. For a system installed on 3390 DASD, the extent for the CF2 parm disk is 1. For a system installed on SCSI volumes, the extent for the CF2 parm disk is 2. After you have made any necessary changes, press F10 to load the module and complete the system IPL.

```

STAND ALONE PROGRAM LOADER: z/VM VERSION 6 RELEASE 2.0

DEVICE NUMBER:  A668      MINIDISK OFFSET:      1  EXTENT:  1
MODULE NAME:    CPLOAD    LOAD ORIGIN:      1000

-----IPL PARAMETERS-----
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=A667 cons=consaddr
-----COMMENTS-----
-----
9= FILELIST  10= LOAD  11= TOGGLE EXTENT/OFFSET
    
```

Testing CMS with LE, the Callable Services Library and the Shared File System

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. VMSERV 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 (VMSERV, VMSERVS, VMSERV, and VMSERVU) specify SAVESEGID CMSFILES. This parameter must be changed to NOSAVESEGID.

xedit *userid dmsparms*

XEDIT the DMSPARMS file for the server.

====> **top**

Start at the top of the file.

====> **set case mixed ignore**

Insure that the search is not case significant.

====> **locate/savesegid**

Locate the SAVESEGID parameter.

====> **change/SAVESEGID *savesegname/NOSAVESEGID***

Change the SAVESEGID *savesegname* parameter to NOSAVESEGID.

====> **file**

Save the updated DMSPARMS file.

define 190 590

Define the production CMS system disk as 590 to allow the user ID to link the test CMS system disk as 190.

define 193 593

Define the production system tools disk as 593 to allow the user ID to link the test system tools disk as 193.

link maint *vrm* 490 190 rr

Link the test CMS system disk as 190.

link maint *vrm* 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).

Testing service for the base components of z/VMz/VM

| z/VM Vv.r.m mm/dd/yy hh:mm

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

access (noprof	Bypass the execution of the PROFILE EXEC.
access 193 b	Access the test system tools disk.
fileserv start	Start the server.
#cp disconnect	Disconnect from the SFS server user ID.

2. Prepare a user ID to be used for testing the new level of CMS.

logon userid	Log on the user ID used to test CMS. The MAINT _{orm} user ID can be used, because it uses the VMSESRVP server for its SFS directories.
define 190 590	Define the production CMS system disk as 590 to allow the user ID to link the test CMS system disk as 190.
define 193 593	Define the production system tools disk as 593 to allow the user ID to link the test system tools disk as 193.
link maint vrm 490 190 rr	Link the test CMS system disk as 190.
link maint vrm 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 Vv.r.m mm/dd/yy hh:mm

3. Test the new level of CMS.
Test the new level of CMS using your installation's method. If you are using SFS, include tests using the SFS server prepared with the new (test) level of CMS. Minimally, this should include XEDITing and saving a file on an SFS directory.
4. Return the user ID used for testing to the old level of CMS.

detach 190	Detach the test CMS system disk.
detach 193	Detach the test system tools disk.
define 590 190	Define the production CMS system disk as 190.
define 593 193	Define the production system tools disk as 193.
ipl 190 clear	IPL the old (production) level of CMS.

| z/VM Vv.r.m mm/dd/yy hh:mm

disconnect	Disconnect from the user ID.
-------------------	------------------------------

5. If the Shared File System was used in the test, return the SFS server to the old level of CMS. Otherwise, skip this task.

logon userid	Log on the SFS server user ID used to test the new level of CMS.
stop	Stop the server.
detach 190	Detach the test CMS system disk.
detach 193	Detach the test system tools disk.
define 590 190	Define the production CMS system disk as 190.
define 593 193	Define the production system tools disk as 193.

Testing service for the base components of z/VMz/VM

```
ipl 190 clear                                IPL the old (production) level of CMS.
| z/VM Vv.r.m mm/dd/yy hh:mm

                ** DO NOT press ENTER! **

access (noprof                               Bypass the execution of the PROFILE EXEC.
access 193 b                                  Access the production system tools disk.
xedit userid dmsparms                       XEDIT the DMSPARMS file for the server.
====> top                                    Start at the top of the file.
====> set case mixed ignore                  Insure that the search is not case significant.
====> locate/nosavesegid                   Locate the NOSAVESEGID parameter.
====> change/NOSAVESEGID/SAVESEGID savesegname
Change the NOSAVESEGID parameter to SAVESEGID
savesegname. If you are using an IBM-supplied server
(VMSERV, VMSERVS, VMSERVR or VMSERVU),
savesegname is CMSFILES.

====> file                                  Save the updated DMSPARMS file.
fileserv start                               Start the server.
#cp disconnect                               Disconnect from the SFS server user ID.
```

6. Log back on to the user ID MAINT_{vr}m to put the service into production.

```
logon MAINTvrm
```

Testing TCP/IP for z/VM Service

All new TCP/IP for z/VM service should be thoroughly tested before it is placed into production. A suggested method for doing this is to temporarily link and access the appropriate test build minidisks ahead of their production counterparts, and then to perform the appropriate tests to confirm that server functions and commands affected by service operate as intended for your installation.

The TCP/IP test build minidisks are owned by the TCP/IP service disk owner (for example, 6VMTCP20, but generalized here as *tcp_ownerid*).

To facilitate the testing of new service that affects TCP/IP server virtual machines, a TCP/IP server profile exit (such as the TCP/IP global profile exit, TCPRUNXT EXEC, supplied in sample form as TCPRUNXT SAMPEXEC) can be used to establish a suitable environment as part of server initialization (SETUP) processing. For information on these exits, see General TCP/IP Server Configuration in *z/VM: TCP/IP Planning and Customization*, as well as the sample content of the TCPRUNXT SAMPEXEC file.

When TCP/IP server and administrative functions are tested, ensure the TCPMAINT 198, *tcp_ownerid* 491, and TCPMAINT 591 minidisks are accessed (in addition to the *tcp_ownerid* 492 and TCPMAINT 592 minidisks) by the TCP/IP service virtual machines and administrative user IDs involved in this activity.

To facilitate the testing of new service that affects TCP/IP client functions (for example, the NETSTAT or FTP commands), only the *tcp_ownerid* 492 and TCPMAINT 592 minidisks need to be accessed (with respect to TCP/IP-specific minidisk requirements).

Testing service for the base components of z/VMz/VM

When new service is tested, consult any applicable documentation (for example, information provided with the APARs that comprise the service) to account for changes that are specific to new or changed function.

Additional testing notes

- Testing procedures also might require TCP/IP services affected by service to be shutdown and restarted, possibly more than once.
- Service to certain TCP/IP components may be relevant to z/VM virtual switch (VSWITCH) support, and may at times necessitate a shutdown and restart of any VSWITCH controller servers that are used by your installation.
- Before you shutdown any TCP/IP or VSWITCH controller servers, ensure that any applicable conditions or guidelines for your installation have been followed.

For information on shutting down TCP/IP servers, see *Stopping TCP/IP Servers* in *z/VM: TCP/IP Planning and Customization*. Note that the TCPMSMGR command can be used to manage the shutdown and initialization of the TCP/IP servers and VSWITCH controllers that are used by your installation. For more information about the TCPMSMGR command, see "Appendix A. TCP/IP Utilities" in the *z/VM TCP/IP Program Directory*.

Appendix C. Testing service for preinstalled components and features

This section provides information on testing a new service level for preinstalled components and features.

RACF Security Server

To test the new service level for RACF you need to perform one of the following procedures:

- If CP and RACF were both serviced, as shown by message VMFSRV1233I in the \$VMFSRV \$MSGLOG, then you must IPL the system using the CF2 PARM disk and you must specify NOAUTOLOG when you are prompted to enter IPL start options. Once the system is IPLed, enter the XAUTOLOG command to autolog the backup RACF server, RACMAINT. For more information, see “IPLing with a test level of the CPLOAD MODULE” on page 36.
- If only RACF was serviced, then you must shutdown the production RACFVM server and bring up the new level of code on the backup RACF server, RACMAINT. To do this, issue the following commands:

```
QUERY SYSOPER (and note the user ID)
SET SYSOPER *
FORCE RACFVM
XAUTOLOG RACMAINT
```

Once you are satisfied with the new level of service, follow these instructions to place the new level into production:

1. Place the service into production by running PUT2PROD. If necessary, see “Steps for placing service into production” on page 12.
2. Start up the production RACFVM server.
 - If CP and RACF were both placed into production, then you must IPL your z/VM system using your primary PARM disk (default CF1) and your normal IPL procedure.
 - If only RACF was placed into production, run the following commands to shutdown the RACMAINT backup server and bring up the RACFVM production server:

```
FORCE RACMAINT
XAUTOLOG RACFVM
SET SYSOPER userid
```

(where *userid* is the *userid* from the previous QUERY SYSOPER command)

Directory Maintenance Facility (DirMaint)

Testing instructions for DirMaint are in “Test the Installation/Service for DirMaint” in *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*

Remote Spooling Communications Subsystem (RSCS)

Testing instructions for RSCS are in “Test the Installation/Service for RSCS” in the RSCS Program Directory.

Appendix D. Local modification example

VMSES/E helps you manage your software inventory by keeping track of the products you have installed, what service is installed for each product, and any local modifications or local service you have installed. This is beneficial because if one type of service impacts another type (for example, a new PTF affects a part you have modified locally) VMSES/E will notify you.

For this reason, IBM recommends that you use the LOCALMOD command whenever you need to install local service or update certain configuration and control files. The following is an example of updating the constants file for the Virtual Machine Resource Manager (VMRM).

VMRM, part of the CMS component, has a configurable constants file, IRMCONS COPY, that defines many values that are needed by VMRM to help you manage the performance of the Linux images on your z/VM system. In this example, you will change the constant "MinRequired" to improve performance for some Linux guest systems.

1. Update the constants file using the LOCALMOD command:

```
localmod cms irmcons copy
VMFLMD2760I LOCALMOD processing started
VMFSET2760I VMFSETUP processing started for SERVP2P CMS
.
.
VMFLMD1301R Local modification L0002 is being created for replacement part
IRMCONS COPY.
Enter (0) to quit; (1) to continue.
If you choose to continue, you will be put into an XEDIT session.
Make your changes and then enter FILE.
```

1
ENTER

At this point you will be placed into an XEDIT session for the IRMCONS COPY file. Locate the constant "minrequired" (by specifying "/minrequired" and pressing Enter) and then type over the value "16384" with your new value (32768, for example). Type "file" on the command line and press Enter. More messages will be displayed by the LOCALMOD command:

```
VMFREP2760I VMFREPL processing started
VMFREP2507I IRMCONS CPYL0002 created on your E-disk from part IRMCONS COPY Q
for use in a VMSES/E environment
VMFREP2760I VMFREPL processing completed successfully
VMFREP2760I VMFREPL processing started
VMFREP2509I The version vector table 6VMCMS20 VVTLCLE will be updated for
the part IRMCONS CPY
VMFREP2760I VMFREPL processing completed successfully
VMFLMD2760I LOCALMOD processing completed successfully
Ready;
```

2. Use the SERVICE command to build VMRM parts with the updated constants file:

```
service cms build
VMFSRV2760I SERVICE PROCESSING STARTED
.
.
VMFBLD2760I VMFBLD PROCESSING STARTED
VMFBLD1851I READING BUILD LISTS
VMFBLD2182I IDENTIFYING NEW BUILD REQUIREMENTS
```

Local Modification Example

```
VMFBLD2182I NEW BUILD REQUIREMENTS IDENTIFIED
VMFBLD2180I THERE ARE 2 BUILD REQUIREMENTS REMAINING
VMFBLD2760I VMFBLD PROCESSING COMPLETED SUCCESSFULLY
VMFBLD2760I VMFBLD PROCESSING STARTED
VMFBLD1851I READING BUILD LISTS
VMFBLD2182I IDENTIFYING NEW BUILD REQUIREMENTS
VMFBLD2182I NO NEW BUILD REQUIREMENTS IDENTIFIED
VMFBLD1851I (1 OF 2) VMFBDCOM PROCESSING DMSBLSRC EXEC Q, TARGET IS BUILD7 493
(K)
VMFBDC2219I PROCESSING OBJECT =.COPY
VMFBLD1851I (1 OF 2) VMFBDCOM COMPLETED WITH RETURN CODE 0
VMFBLD1851I (2 OF 2) VMFBDCOM PROCESSING DMSBL493 EXEC Q, TARGET IS BUILD7 493
(K)
VMFBDC2219I PROCESSING OBJECT IRMCONS.COPY
VMFBLD1851I (2 OF 2) VMFBDCOM COMPLETED WITH RETURN CODE 0
VMFBLD2180I THERE ARE 0 BUILD REQUIREMENTS REMAINING
VMFBLD2760I VMFBLD PROCESSING COMPLETED SUCCESSFULLY
.
.
VMFSRV1233I THE FOLLOWING PRODUCTS HAVE BEEN SERVICED.
VMFSRV1233I CMS
VMFSRV2760I SERVICE PROCESSING COMPLETED SUCCESSFULLY
Ready;
```

3. Use the PUT2PROD command to put the updated parts into production:

put2prod

Note: If you are running an SSI cluster, on each member of the SSI cluster you must log on to MAINT_{vrn} and run the PUT2PROD command.

Appendix E. Typical local modifications

Local modifications can be useful in helping you perform some systems management functions. Two common functions are discussed in this section.

Modifying the CMSINST segment

CMS improves system performance and storage usage by placing heavily used EXECs and MODULES in the CMS installation segment, CMSINST. You can change the contents of the CMSINST segment (for example, you can add or delete an application EXEC) by identifying 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 must be updated in order to make 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.
- DMSBINS EXEC is the CMSINST segment build list. It includes a :BLDREQ record for each of the EXECs in CMSINST that belongs 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 DMSBINS 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 physical segment, which contains the CMSINST logical segment, is large enough for the additional EXECs. You can determine how much space is available by looking at the entry for INSTSEG in the SYSTEM SEGID file, or by looking at the INSTSEG PSEGMAP, if it exists. The INSTSEG PSEGMAP tells you how many bytes are unused in the physical segment. Use the VMSES/E segment mapping tool, VMFSGMAP, to increase the range defined for INSTSEG if there is not enough free space to hold the EXECs you are adding to the CMSINST segment.

Also, if an EXEC you are adding is not on a disk normally 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. For more information about the VMFSGMAP tool, see *z/VM: VMSES/E Introduction and Reference*.

- a. If changes are required to the INSTSEG segment invoke the segment mapping tool (VMFSGMAP) to change the segment definition.

vmfsgmap segbld esasegs segblist

This command will display a panel containing a map of the segments on your system. Place the cursor on the INSTSEG segment and select PF4 to display the change segment definition panel.

Typical Local Modifications

- 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. Use the LOCALMOD command to update CMSINST LSEG.
 - a. Enter the following command:

localmod cms cmsinst lseg

```
VMFLMD1301R Local modification L0004 is being created for
replacement part CMSINST LSEG.
Enter (0) to quit; (1) to continue.
If you choose to continue, you will be put into an
XEDIT session.
Make your changes and then enter FILE.
```

1

The LOCALMOD command will access the required CMS disks and create a new local modification file from the highest level part available and then present you with an XEDIT session. Respond "1" to the XEDIT prompt to continue.

- b. Once you are in the XEDIT session:

```
====> input EXEC C89 EXEC *
====> file
VMFREPL2760I VMFREPL processing started
VMFREPL2507I CMSINST SEGL0004 created on your E-disk from part
CMSINST LSEG Q for use in a VMSES/E environment
VMFREPL2760I VMFREPL processing completed successfully
VMFREPL2760I VMFREPL processing started
VMFREPL2509I The version vector table 6VMCMS20 VVTLCCL E will
be updated for the part CMSINST SEG
VMFREPL2760I VMFREPL processing completed successfully
VMFLMD2760I LOCALMOD processing completed successfully
Ready;
```

Add the entry for the C89 EXEC. The LOCALMOD command completes for CMSINST LSEG.

3. Update the DMSSBINS EXEC to add the build requirement for the C89 EXEC:

Since the C89 EXEC belongs to the CMS component, the DMSSBINS EXEC must be modified to add a :BLDREQ record it. This ensures that when the C89 EXEC is serviced, the CMSINST saved segment also gets built. If you are adding an EXEC that does not belong to the CMS component, skip this step and continue to step 4 on page 47.

 - a. Create the local modification file for the DMSSBINS EXEC:

localmod cms dmssbins exec

```
VMFLMD1301R Local modification L0005 is being created for update
part DMSSBINS$EXEC.
Enter (0) to quit; (1) to continue.
If you choose to continue, you will be put into
an XEDIT session.
Make your changes and then enter FILE.
```

1

The LOCALMOD command will access the required CMS disks and create a new local modification file from the highest level part available and then present you with an XEDIT session. Respond "1" to the XEDIT prompt to continue.

- b. Once you are in the XEDIT session:

```
====> locate /:BLDREQ./
```

Locate the beginning of the :BLDREQ records and add the build requisite for the C89 EXEC. DMSBLCRX is the build list that contains the C89 EXEC.

```
====> input          DMSBLCRX.C89.EXEC
====> file = = e
VMFEXU2760I VMFEXUPD processing started
DMSUPD178I Updating DMSSBINS $EXEC A1
DMSUPD178I Applying DMSSBINS UPL0005 E1
VMFSIP2509I The version vector table 6VMCMS20 VVTLCLE will be
             updated for the part DMSSBINS EXC using the
             information in AUX file DMSSBINS AUXLCL
VMFEXU2507I DMSSBINS EXCL0005 created on your E-disk for use
             in a VMSES/E environment
VMFEXU2760I VMFEXUPD processing completed successfully
VMFLMD2760I LOCALMOD processing completed successfully
Ready;
```

File the change to DMSSBINS \$EXEC. The change now exists on the LOCALMOD disk. The LOCALMOD command completes for the DMSSBINS EXEC.

4. Rebuild the affected objects and place your local modification into production.
 - a. First issue the following command to rebuild CMSINST LSEG and DMSSBINS EXEC.

```
service cms build
```

- b. Use the PUT2PROD command to rebuild the INSTSEG saved segment which contains the CMSINST logical saved segment.

```
put2prod
```

Note: If you are running an SSI cluster, on each member of the SSI cluster you must log on to `MAINTvrm` and run the PUT2PROD command.

Updating the CP load list

This section uses the LOCALMOD command 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 information on 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, you must add it to the HCPMDLAT

Typical Local Modifications

MACRO (or equivalent) 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, use these instructions. This involves:

- Reviewing the guidelines for using update files
- Creating a local modification for HCPMDLAT MACRO
- Issuing the GENCPBLS command to create the CP load list
- Updating the HCPGPI MACLIB, which contains HCPMDLAT MACRO
- Rebuilding the CP nucleus

Steps for updating the CP load list

Perform the following steps to update the CP load list.

1. Review Guidelines for using update files.
2. Modify HCPMDLAT MACRO using the LOCALMOD command. The HCPMDLAT MACRO is source-maintained, so the High Level Assembler is required in order to make changes to this macro. Make sure that you have installed the High Level Assembler on your system and placed the assembler on a disk that is always accessed (for example, the MAINT 19E disk) or will be accessed when VMSES/E commands are executed.

```
| localmod cp hcpmdlat macro
VMFLMD1301R Local modification L0001 is being created
             for update part HCPMDLAT MACRO.
             Enter (0) to quit; (1) to continue ;
             (2) to create a dummy or unchanged part
             If you choose to continue, you will be put
             into an XEDIT session.
             Make your changes and then enter FILE.
```

1 ENTER

Use the LOCALMOD command to set up the correct environment and create an update file. The LOCALMOD command will prompt you to either make modifications to the HCPMDLAT MACRO (Option 1), or, if you only need to pick up a xxxMDLAT MACRO, create a dummy update to the HCPMDLAT MACRO (Option 2).

Typical Local Modifications

```
PRT FILE 0126 SENT FROM MAINT620 PRT WAS 0126 RECS
      2090 COPY 001 A NOHOLD NOKEEP
VMFGCB2760I GENCPBLS processing started
VMFGCB2760I HCPMDLAT MACRO processing started
DMSUPD178I Updating HCPMDLAT MACRO A1
DMSUPD178I Applying HCPMDLAT UPL0001 E1
VMFSIP2509I The version vector table 6VMCPR20 VVTLCL E
             will be updated for HCPMDLAT AUXLCL the part
             HCPMDLAT MACRO using the information in AUX file
VMFGCB2760I HCPLDL ASSEMBLE processing started
VMFGCB2409I File HCPLDL AUXLCL will be created on 2C4 (E)
DMSUPD178I Updating HCPLDL ASSEMBLE A1
DMSUPD178I Applying HCPLDL UPL0001 E1
DMSOPN002E File RACF MACLIB * not found
VMFGCB1907I Assembling HCPLDL
      Assembler Done LDL No Statements Flagged

VMFGCB2507I CPLOAD EXCL0001 created on your E-disk for use in a
             VMSES/E environment
VMFGCB2760I GENCPBLS processing completed successfully
VMFLMD2760I LOCALMOD processing completed successfully
Ready;
```

You will be placed in an XEDIT session. Make your changes to the displayed file. Enter "file" on the command line when you are done. LOCALMOD processing will continue with the GENCPBLS command. This command will update the necessary VMSES/E control files and generate an update file for the CPLOAD EXEC

3. Rebuild HCPGPI MACLIB and the CP nucleus with the SERVICE command:

```
service cp build
```

4. Use the PUT2PROD command to put the rebuilt parts into production:

```
put2prod
```

Note: If you are running an SSI cluster, on each system in the SSI cluster you must log on to MAINT $_{vrm}$ and run the PUT2PROD command.

Typical Local Modifications

Notices

This information was developed for products and services offered in the U.S.A.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not 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, NY 10504-1785
U.S.A.

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

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

The licensed program described in this 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. The

sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see the IBM Online Privacy Policy at <http://www.ibm.com/privacy> and the IBM Online Privacy Statement at <http://www.ibm.com/privacy/details>, in particular the section entitled "Cookies, Web Beacons and Other Technologies", and the IBM Software Products and Software-as-a-Service Privacy Statement at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States (www.ibm.com/legal/us/en/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Glossary

For a list of z/VM terms and their definitions, see *z/VM: Glossary*.

The z/VM glossary is also available through the online z/VM HELP Facility, if HELP files are installed on your z/VM system. For example, to display the definition of the term “dedicated device”, issue the following HELP command:
`help glossary dedicated device`

While you are in the glossary help file, you can do additional searches:

- To display the definition of a new term, type a new HELP command on the command line:

```
help glossary newterm
```

This command opens a new help file inside the previous help file. You can repeat this process many times. The status area in the lower right corner of the screen shows how many help files you have open. To close the current file, press the Quit key (PF3/F3). To exit from the HELP Facility, press the Return key (PF4/F4).

- To search for a word, phrase, or character string, type it on the command line and press the Clocate key (PF5/F5). To find other occurrences, press the key multiple times.

The Clocate function searches from the current location to the end of the file. It does not wrap. To search the whole file, press the Top key (PF2/F2) to go to the top of the file before using Clocate.

Bibliography

See the following publications for additional information about z/VM. For abstracts of the z/VM publications, see *z/VM: General Information*, GC24-6193

Where to get z/VM information

z/VM product information is available from the following sources:

- z/VM V6.3 Information Center (publib.boulder.ibm.com/infocenter/zvm/v6r3/)
- IBM: z/VM Internet Library (www.ibm.com/vm/library/)
- IBM Publications Center (www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)
- *IBM Online Library: z/VM Collection*, SK5T-7054

z/VM base library

Overview

- *z/VM: General Information*, GC24-6193
- *z/VM: Glossary*, GC24-6195
- *z/VM: License Information*, GC24-6200

Installation, migration, and service

- *z/VM: Installation Guide*, GC24-6246
- *z/VM: Migration Guide*, GC24-6201
- *z/VM: Service Guide*, GC24-6247
- *z/VM: VMSES/E Introduction and Reference*, GC24-6243

Planning and administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6167
- *z/VM: CMS Planning and Administration*, SC24-6171
- *z/VM: Connectivity*, SC24-6174
- *z/VM: CP Planning and Administration*, SC24-6178
- *z/VM: Getting Started with Linux on System z*, SC24-6194
- *z/VM: Group Control System*, SC24-6196
- *z/VM: I/O Configuration*, SC24-6198

- *z/VM: Running Guest Operating Systems*, SC24-6228
- *z/VM: Saved Segments Planning and Administration*, SC24-6229
- *z/VM: Secure Configuration Guide*, SC24-6230
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6236
- *z/VM: TCP/IP Planning and Customization*, SC24-6238
- *z/OS and z/VM: Hardware Configuration Manager User's Guide*, SC33-7989

Customization and tuning

- *z/VM: CP Exit Customization*, SC24-6176
- *z/VM: Performance*, SC24-6208

Operation and use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6166
- *z/VM: CMS Pipelines Reference*, SC24-6169
- *z/VM: CMS Pipelines User's Guide*, SC24-6170
- *z/VM: CMS Primer*, SC24-6172
- *z/VM: CMS User's Guide*, SC24-6173
- *z/VM: CP Commands and Utilities Reference*, SC24-6175
- *z/VM: System Operation*, SC24-6233
- *z/VM: TCP/IP User's Guide*, SC24-6240
- *z/VM: Virtual Machine Operation*, SC24-6241
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6244
- *z/VM: XEDIT User's Guide*, SC24-6245
- *CMS/TSO Pipelines: Author's Edition*, SL26-0018

Application programming

- *z/VM: CMS Application Development Guide*, SC24-6162
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6163
- *z/VM: CMS Application Multitasking*, SC24-6164
- *z/VM: CMS Callable Services Reference*, SC24-6165
- *z/VM: CMS Macros and Functions Reference*, SC24-6168
- *z/VM: CP Programming Services*, SC24-6179
- *z/VM: CPI Communications User's Guide*, SC24-6180

- z/VM: *Enterprise Systems Architecture/Extended Configuration Principles of Operation*, SC24-6192
- z/VM: *Language Environment User's Guide*, SC24-6199
- z/VM: *OpenExtensions Advanced Application Programming Tools*, SC24-6202
- z/VM: *OpenExtensions Callable Services Reference*, SC24-6203
- z/VM: *OpenExtensions Commands Reference*, SC24-6204
- z/VM: *OpenExtensions POSIX Conformance Document*, GC24-6205
- z/VM: *OpenExtensions User's Guide*, SC24-6206
- z/VM: *Program Management Binder for CMS*, SC24-6211
- z/VM: *Reusable Server Kernel Programmer's Guide and Reference*, SC24-6220
- z/VM: *REXX/VM Reference*, SC24-6221
- z/VM: *REXX/VM User's Guide*, SC24-6222
- z/VM: *Systems Management Application Programming*, SC24-6234
- z/VM: *TCP/IP Programmer's Reference*, SC24-6239
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- z/OS: *IBM Tivoli Directory Server Plug-in Reference for z/OS*, SA76-0148
- 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 Interlanguage Communication 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*, GC24-6161
- z/VM: *CP Messages and Codes*, GC24-6177

- z/VM: *Diagnosis Guide*, GC24-6187
- z/VM: *Dump Viewing Facility*, GC24-6191
- z/VM: *Other Components Messages and Codes*, GC24-6207
- z/VM: *TCP/IP Diagnosis Guide*, GC24-6235
- z/VM: *TCP/IP Messages and Codes*, GC24-6237
- z/VM: *VM Dump Tool*, GC24-6242
- z/OS and z/VM: *Hardware Configuration Definition Messages*, SC33-7986

z/VM facilities and features

Data Facility Storage Management Subsystem for VM

- z/VM: *DFSMS/VM Customization*, SC24-6181
- z/VM: *DFSMS/VM Diagnosis Guide*, GC24-6182
- z/VM: *DFSMS/VM Messages and Codes*, GC24-6183
- z/VM: *DFSMS/VM Planning Guide*, SC24-6184
- z/VM: *DFSMS/VM Removable Media Services*, SC24-6185
- z/VM: *DFSMS/VM Storage Administration*, SC24-6186

Directory Maintenance Facility for z/VM

- z/VM: *Directory Maintenance Facility Commands Reference*, SC24-6188
- z/VM: *Directory Maintenance Facility Messages*, GC24-6189
- z/VM: *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190

Open Systems Adapter/Support Facility

- zEnterprise System, System z10, 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
- System z: *Open Systems Adapter-Express Integrated Console Controller 3215 Support*, SA23-2247
- System z10: *Open Systems Adapter-Express3 Integrated Console Controller Dual-Port User's Guide*, SA23-2266

Performance Toolkit for VM

- z/VM: *Performance Toolkit Guide*, SC24-6209
- z/VM: *Performance Toolkit Reference*, SC24-6210

RACF Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide, SC24-6212*
- *z/VM: RACF Security Server Command Language Reference, SC24-6213*
- *z/VM: RACF Security Server Diagnosis Guide, GC24-6214*
- *z/VM: RACF Security Server General User's Guide, SC24-6215*
- *z/VM: RACF Security Server Macros and Interfaces, SC24-6216*
- *z/VM: RACF Security Server Messages and Codes, GC24-6217*
- *z/VM: RACF Security Server Security Administrator's Guide, SC24-6218*
- *z/VM: RACF Security Server System Programmer's Guide, SC24-6219*
- *z/VM: Security Server RACROUTE Macro Reference, SC24-6231*

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis, GC24-6223*
- *z/VM: RSCS Networking Exit Customization, SC24-6224*
- *z/VM: RSCS Networking Messages and Codes, GC24-6225*
- *z/VM: RSCS Networking Operation and Use, SC24-6226*
- *z/VM: RSCS Networking Planning and Configuration, SC24-6227*
- *Network Job Entry: Formats and Protocols, SA22-7539*

Prerequisite products

Device Support Facilities

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

Environmental Record Editing and Printing Program

- *Environmental Record Editing and Printing Program (EREP): Reference, GC35-0152*
- *Environmental Record Editing and Printing Program (EREP): User's Guide, GC35-0151*

Index

A

alternate
 apply disk 5
APAR descriptions 5
APPLY disks 5
applying service
 overview 3
AUX
 files 3

B

BASE disks 5
BUILD disks 5

C

changing, CP load list 45
CMS nucleus 23
CMSINST
 CMSINST LSEG 45
 DMSSBINS EXEC 45
 local modification 45
 updating 45
component(s)
 order of service 4
COR service
 main tasks 2
corequisite publications 1
CP load list, updating 45
CP nucleus 23

D

DELTA disk 5
Directory Maintenance Facility (DirMaint) 41
disk(s)
 used for servicing 2
disks used to service z/VM 5
DMSNGP ASSEMBLE 5

F

files
 COR VMELEDOC 33
 COR01 MEDIAMAP 33
 DMSPARMS 37
 nsdocenvfn SERVLINK 33
 SERVICE DOCUMENT 33

G

GCS nucleus 23

H

HCPMDLAT MACRO
 CP load list 45

HCPMDLAT MACRO (*continued*)
 updating xxxMDLAT MACRO 45
HELP saved segment
 example, CMSINST 45

L

load list, updating CP load list 45
local
 disk 5
 example, CMSINST 45
 sample disk 5
local modification example 43
local service
 CP load list 45
 CPLOAD EXEC 45
 example, CMSINST 45
 guidelines for update files 16
 overview 3, 15
 reworking overview 15
LOCALMOD
 using 18
LOCALMOD disk 5
LOCALSAM disk 5
LOGO CONFIG 5

M

merge processing
 overview 3

N

non-VMSES envelopes, receiving 33

O

objects 5
order of service 4
out-of-component requisites 4
overview of procedures 1

P

procedures, overview 1
production
 delta disk 5
production, placing serviced components into
 overview 3
 procedure 12
PTF-numbered parts 5

R

RACF Security Server 41
receiving service
 non-VMSES envelopes 33
 overview 3

- reference service information 1
- Remote Spooling Communications Subsystem (RSCS) 41
- removing service
 - backing out a local modification 25
 - backing out a single PTF 25
 - local modification 25
 - overview 25
 - service level 26
 - service levels 26
 - single PTF 25
- rework local service 15
- RSU (Recommended Service Upgrade)
 - checking level 8
 - overview 1

S

- saved systems 23
- segments 23
- sequence of servicing components 4
- service
 - disks used 5
 - installing 7
 - main tasks 2
 - ordering 8
 - types 1
- service procedures 1
- service, covered components, features, and products 7
- Stand-alone builds 23
- SYSTEM CONFIG 5
- SYSTEM disks 5

T

- TASK disks 5
- tasks
 - CP load list, updating 48
 - Setting up VMSES/E Licensed Products to use the SERVICE
 - steps for 29
- testing service
 - base components 35
 - CMS 37
 - CP 36
 - Directory Maintenance Facility (DirMaint) 41
 - LE 37
 - preinstalled components 41
 - RACF Security Server 41
 - Remote Spooling Communications Subsystem (RSCS) 41
 - TCP/IP 39
- types of service 1

U

- update files 16

V

- VMFMRDSK EXEC 3
- VMFPLCD command
 - using the command> 33
- VMFREM EXEC 25
- VMFSGMAP
 - example, CMSINST 45
- VMLIB 37

- VMSES/E command information 1

X

- xxxMDLAT MACRO
 - CP load list 45
 - updating xxxMDLAT MACRO 45



Product Number: 5741-A07

Printed in USA

GC24-6247-01

