

z/VM
7.4

Service Guide



Note:

Before you use this information and the product it supports, read the information in [“Notices” on page 85.](#)

This edition applies to version 7, release 4 of IBM® z/VM® (product number 5741-A09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-06-11

© **Copyright International Business Machines Corporation 1991, 2025.**

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

Contents

Figures.....	vii
Tables.....	ix
About this document.....	xi
Intended audience.....	xi
Conventions and terminology.....	xi
Using the online HELP facility.....	xii
Where to find more information.....	xii
Links to other documents and websites.....	xiii
How to provide feedback to IBM.....	xv
Summary of changes for z/VM: Service Guide.....	xvii
GC24-6325-74, z/VM 7.4 (June 2025).....	xvii
GC24-6325-74, z/VM 7.4 (March 2025).....	xvii
GC24-6325-74, z/VM 7.4 (January 2025).....	xvii
GC24-6325-74, z/VM 7.4 (September 2024).....	xviii
GC24-6325-73, z/VM 7.3 (June 2025).....	xix
GC24-6325-73, z/VM 7.3 (July 2024).....	xix
GC24-6325-73, z/VM 7.3 (March 2024).....	xix
GC24-6325-73, z/VM 7.3 (December 2023).....	xix
GC24-6325-73, z/VM 7.3 (September 2023).....	xix
GC24-6325-73, z/VM 7.3 (May 2023).....	xix
GC24-6325-73, z/VM 7.3 (September 2022).....	xx
Part 1. Servicing traditional z/VM systems and SSI members.....	1
Chapter 1. Servicing your system.....	3
Overview of the service procedure.....	3
Merging service.....	4
Receiving service.....	4
Applying service.....	4
Reworking local service (if applicable).....	5
Building new levels.....	5
Testing serviced components.....	5
Placing the serviced components into production.....	5
Component dependencies and service order.....	5
Out-of-component requisites.....	6
Product parameter file.....	6
Disks used to service z/VM.....	6
Chapter 2. Ordering and retrieving service.....	9
Retrieving service.....	9
Using GETSHOPZ to obtain service.....	11
Chapter 3. Installing preventive or corrective service.....	17
Requirements and considerations.....	17
Steps to install preventive or corrective service.....	18

Steps for placing service into production.....	20
Chapter 4. Applying local service and local modifications.....	23
Overview of the local modification process.....	23
Applying or creating a local modification.....	25
Reworking a local modification.....	26
Applying local service.....	27
Chapter 5. Running standalone builds.....	29
Rebuilding the CP, CMS, or GCS nucleus.....	29
Rebuilding segments.....	29
Rebuilding the CMS saved system.....	29
Chapter 6. Removing service.....	31
Removing a single PTF.....	31
Removing a local modification.....	31
Restoring removed service.....	32
Removing service levels.....	32
Chapter 7. Setting up VMSES/E licensed products to use the SERVICE command.....	35
Part 2. Using z/VM Centralized Service Management to service z/VM systems.....	37
Chapter 8. An overview of z/VM Centralized Service Management (z/VM CSM).....	39
A high-level flow of z/VM CSM.....	39
System requirements for z/VM CSM.....	40
Software requirements for z/VM CSM.....	41
Chapter 9. Planning for and configuring z/VM CSM.....	43
Overall system configuration changes.....	43
SFS configuration changes.....	43
TCP/IP configuration changes.....	44
Using nicknames for multiple systems.....	45
Chapter 10. Setting up z/VM CSM.....	47
The SERVMGR command.....	47
Initialization and system setup.....	48
Adding service levels.....	48
Adding managed systems.....	49
Packaging and sending service levels.....	50
Putting service into production.....	51
z/VM CSM service for external security managers	52
Chapter 11. Managing a z/VM CSM environment.....	53
Querying systems and service level information.....	53
Systems.....	53
Service levels.....	55
Managing service levels and their content.....	56
Maintaining managed system information.....	56
Special service management tasks.....	57
Deleting a version/release/modification level.....	57
Reverting from z/VM CSM to traditional service management.....	58
Appendix A. z/VM service environment variables.....	59
Appendix B. Service-related utilities.....	61
GETSHOPZ EXEC.....	62

Appendix C. Receiving service for non-VMSES/E envelopes.....	67
Appendix D. Testing service for the base components of z/VM.....	69
Testing components with new service.....	69
IPLing with a test level of the CPLOAD MODULE.....	70
Testing CMS with the Callable Services Library, Shared File System, and REXX/VM.....	71
Testing TCP/IP for z/VM service.....	74
Additional testing notes.....	74
Appendix E. Testing service for preinstalled components and features.....	75
RACF Security Server.....	75
Directory Maintenance Facility (DirMaint).....	76
Remote Spooling Communications Subsystem (RSCS).....	76
Appendix F. Local modification example.....	77
Appendix G. Typical local modifications.....	79
Modifying the CMSINST segment.....	79
Updating the CP load list.....	81
Steps for updating the CP load list.....	82
Notices.....	85
Trademarks.....	86
Terms and conditions for product documentation.....	86
IBM online privacy statement.....	87
Bibliography.....	89
Where to Get z/VM Information.....	89
z/VM Base Library.....	89
z/VM Facilities and Features.....	90
Prerequisite Products.....	92
Related Products.....	92
Index.....	93

Figures

I 1. Disks used when installing service..... 4

Tables

1. Preinstalled components, features, and products, in preferred service order17

2. Test build and production disks for preinstalled components and features..... 69

About this document

This document describes how to service IBM z/VM 7.4 and includes information about z/VM Centralized Service Management (z/VM CSM).

Information about the preventive and corrective service procedures for all components, local service and modifications, and reworking local service is also included, as well as supplemental information and procedures.

Intended audience

This information is intended for individuals responsible for maintaining and servicing one or more z/VM systems. It is assumed that you understand the concepts of z/VM service and are familiar with the syntax and usage of the z/VM service tools. For more information about the z/VM service concepts and tools, see *z/VM: VMSES/E Introduction and Reference*.

Conventions and terminology

In this book, the term MAINT`vrn`, where `vrn` 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 tapeaddr * 181
```

```
TAPE tapeaddr ATTACHED TO userID 181  
Ready;
```

tapeaddr

is the address of the tape drive(s) where the z/VM System DDR tapes will be mounted.

userID

is the first-level user ID logged on to in the previous substep.

- Normal font indicates system responses and requests.

Example: The following shows a system response:

```
IUGIPX8475I Final selections display  
The products you selected to load to minidisk are:  
NONE
```

```
The products you selected to load to SFS are:  
VM PERFTK RACF DIRM RSCS ICKDSF TCPIP  
:
```

- 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 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 vmstes menu
```

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

```
help vmstes 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 [z/VM: CMS User's Guide](#). To display the main HELP Task Menu, enter:

```
help
```

For more information about the HELP command, see [z/VM: CMS Commands and Utilities Reference](#) or enter:

```
help cms help
```

Where to find more information

For information about related documents, see the [“Bibliography”](#) on page 89.

Links to other documents and websites

The PDF version of this document contains links to other documents and websites. 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 website 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 provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See [How to send feedback to IBM](#) for additional information.

Summary of changes for z/VM: Service Guide

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line (|) to the left of the change.

GC24-6325-74, z/VM 7.4 (June 2025)

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.4.

[VM66840] z/VM service environment variables

With the PTF for APAR VM66840, environment variables are established to provide the ability to easily determine the latest service that is installed on a z/VM 7.4 system. z/VM service environment variables allow administrators or automation to view the linear service level from CP, CMS, or a Linux® guest.

The following topic is new:

- [Appendix A, “z/VM service environment variables,” on page 59](#)

The following topic is updated:

- [Chapter 2, “Ordering and retrieving service,” on page 9](#)

For more information, see:

- [z/VM: VMSES/E Introduction and Reference](#)
- [z/VM: Other Components Messages and Codes](#)

GC24-6325-74, z/VM 7.4 (March 2025)

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.4.

[VM66834] GETSHOPZ ServiceLink support

With the PTF for APAR VM66834, z/VM 7.4 provides GETSHOPZ support for direct-to-host transfer of service packages from IBM ServiceLink. Clients currently use the GETSHOPZ utility to securely transfer z/VM service packages directly from Shopz. This support provides the same ability for ServiceLink orders.

The following topics are updated:

- [“Retrieving service” on page 9](#)
- [“Using GETSHOPZ to obtain service” on page 11](#)
- [“GETSHOPZ EXEC” on page 62](#)

GC24-6325-74, z/VM 7.4 (January 2025)

This edition includes terminology, maintenance, and editorial changes.

The following topic is updated:

- [Chapter 2, “Ordering and retrieving service,” on page 9](#)

GC24-6325-74, z/VM 7.4 (September 2024)

This edition supports the general availability of z/VM 7.4. Note that the publication number suffix (-74) indicates the z/VM release to which this edition applies.

Linear service

z/VM 7.4 introduces linear service to the product for components at the 740 function level. Corrective service updates — in the form of fix packs, hot fixes, and hardware support — and new functions in feature packs are released in service stream PTFs. The latest PTF identifies the requisites of all fixes and features for a component up to that point. For more information about linear service, including schedules and the types of APARs, see:

Introducing z/VM Linear Service (<https://www.vm.ibm.com/service/linear.html>)

New format for z/VM product service level

Starting with z/VM 7.4, the z/VM product service level consists of a new format — *ffxx* — where:

- *ff* indicates the latest feature pack number
- *xx* indicates the fix pack number for the latest feature pack.

In addition, a new identifier — LVL — denotes preventive service — LVL-0001, for example.

The following topics are updated:

- Chapter 2, “Ordering and retrieving service,” on page 9
- “Adding service levels” on page 48

Integration of HCD and HCM for z/VM into the CP component base

The functional executables of the Hardware Configuration Definition and Hardware Configuration Manager for z/VM (HCD and HCM for z/VM) are now shipped as part of the CP component of z/VM 7.4. Service support is provided with the CP component of z/VM.

Help files for HCD are not included with the z/VM system. For information about HCD, see the following:

- Using HCM and HCD in *z/VM: I/O Configuration*
- *z/OS and z/VM: Hardware Configuration Manager User's Guide* (https://www.ibm.com/docs/en/SSLTBW_3.1.0/pdf/eequ100_v3r1.pdf)
- *z/OS and z/VM: Hardware Configuration Definition Messages* (https://www.ibm.com/docs/en/SSLTBW_3.1.0/pdf/cbdl100_v3r1.pdf)

Removal of physical DVDs as a product distribution medium

DVD-R is no longer a supported system image distribution medium. The replacement for installing from a physical DVD on an HMC is to use a USB flash drive. An electronic image in ISO format will be provided. Customers can copy the contents of the electronic image to a USB flash drive by following instructions provided with their order. Only USB flash drives supported by the HMC can be used.

This satisfies the Statement of Direction from the z/VM 7.3 product announcement.

Miscellaneous updates for z/VM 7.4

- Starting with z/VM 7.4, Preventive Service Planning (PSP) buckets will no longer be created for product releases.

The following topic is updated: “Steps to install preventive or corrective service” on page 18.

GC24-6325-73, z/VM 7.3 (June 2025)

This edition includes terminology, maintenance, and editorial changes.

The following topic is updated:

- [“RACF Security Server” on page 75](#)

GC24-6325-73, z/VM 7.3 (July 2024)

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.3.

[VM66774] GETSHOPZ enhancements

With the PTF for APAR VM66774, z/VM 7.3 provides enhancements to the GETSHOPZ utility.

The following topics are updated:

- [“Using GETSHOPZ to obtain service” on page 11](#)
- [“GETSHOPZ EXEC” on page 62](#)

GC24-6325-73, z/VM 7.3 (March 2024)

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.3.

[VM66732] Digital Signature Verification of z/VM Service Packages

With the PTF for APAR VM66732, z/VM 7.3 uses public-key cryptography to verify the authenticity and integrity of signed service packages.

The following topics are updated:

- [“Using GETSHOPZ to obtain service” on page 11](#)
- [“GETSHOPZ EXEC” on page 62](#)

GC24-6325-73, z/VM 7.3 (December 2023)

This edition includes terminology, maintenance, and editorial changes.

GC24-6325-73, z/VM 7.3 (September 2023)

This edition supports product changes that were provided or announced after the general availability of z/VM 7.3.

[PH56199, VM66698] System SSL z/OS 2.5 Equivalence

With the PTFs for APARs PH56199 (TCP/IP) and VM66698 (LE), z/VM 7.3 provides an update to the cryptographic services library, which includes certificate diagnostic enhancements and improved algorithmic support and allows for enablement of TLS 1.3, for secure connectivity to the z/VM platform.

The following topic is updated:

- [“Using GETSHOPZ to obtain service” on page 11](#)

GC24-6325-73, z/VM 7.3 (May 2023)

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.3.

[VM66453, VM66457, PH51239] z/VM Centralized Service Management Use of CMS Keyvault

With the PTFs for APARs VM66453 (CMS), VM66457 (VMSES/E), and PH51239 (TCP/IP), z/VM 7.3 provides support for a CMS password/key management utility called KEYVAULT, which allows applications to securely store and retrieve user ID keys (logon passwords). z/VM Centralized Service Management (z/VM CSM) and the TCP/IP FTP client are updated to use the new KEYVAULT utility for automated remote host login procedures.

The following topic is updated:

- [“The SERVMGR command” on page 47](#)

GC24-6325-73, z/VM 7.3 (September 2022)

This edition includes changes to support the general availability of z/VM 7.3. Note that the publication number suffix (-73) indicates the z/VM release to which this edition applies.

z/VM Centralized Service Management enhancements

z/VM Centralized Service Management (z/VM CSM) includes the following usability enhancements:

- QUERY processing for service levels now allows queries that are specific to individual components in a service level.
- Wildcard support is added to the SERVMGR SRVLVL QUERY command.
- The ability to query z/VM CSM managed systems for current PUT2PROD status is added.
- Additional details are provided when querying local modifications.

The following topic is updated:

- [“Querying systems and service level information” on page 53](#) > [“Systems” on page 53](#)

Language Environment upgrade

The z/VM Language Environment runtime libraries have been upgraded to z/OS 2.5 equivalence.

The following topics are updated:

- [Chapter 7, “Setting up VMSES/E licensed products to use the SERVICE command,” on page 35](#)
- [“Testing components with new service” on page 69](#)

Additional z/VM 7.3 changes

The following topics are new:

- [“Upgrading a z/VM CSM-managed system to a new release of z/VM” on page 57](#)
- [“Deleting a version/release/modification level” on page 57](#)
- [“Reverting from z/VM CSM to traditional service management” on page 58](#)

The following topics are updated:

- [Chapter 2, “Ordering and retrieving service,” on page 9](#)
- [“Retrieving service” on page 9](#)
- [“Using GETSHOPZ to obtain service” on page 11](#)
- [“Steps to install preventive or corrective service” on page 18](#)
- [“System requirements for z/VM CSM” on page 40](#)
- [“Software requirements for z/VM CSM” on page 41](#)
- [“Overall system configuration changes” on page 43](#)

- [“Adding service levels” on page 48](#)
- [“Packaging and sending service levels” on page 50](#)

Part 1. Servicing traditional z/VM systems and SSI members

This part includes information about servicing traditional z/VM systems and systems in an SSI.

Chapter 1. Servicing your system

This topic 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 (RSU) service
- Corrective (COR) service
- Local service
- Local modifications.

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

z/VM 7.4 introduces linear service to the product for components at the 740 function level. Corrective service updates — in the form of fix packs, hot fixes, and hardware support — and new functions in feature packs are released in service stream PTFs. The latest PTF identifies the requisites of all fixes and features for a component up to that point. For more information about linear service, including schedules and the types of APARs, see:

Introducing z/VM Linear Service (<https://www.vm.ibm.com/service/linear.html>)

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 in a preventive (RSU) or corrective (COR) service envelope. 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.

Notes:

1. Preventive (RSU) and corrective (COR) service updates are available electronically as service envelopes (with a file type of SERVLINK).

For more information about ordering and receiving service, see:

IBM: z/VM Support Resources (<https://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 receives, applies, identifies, and builds 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 on page 4 shows the flow of information from disk to disk when you install service for a component.

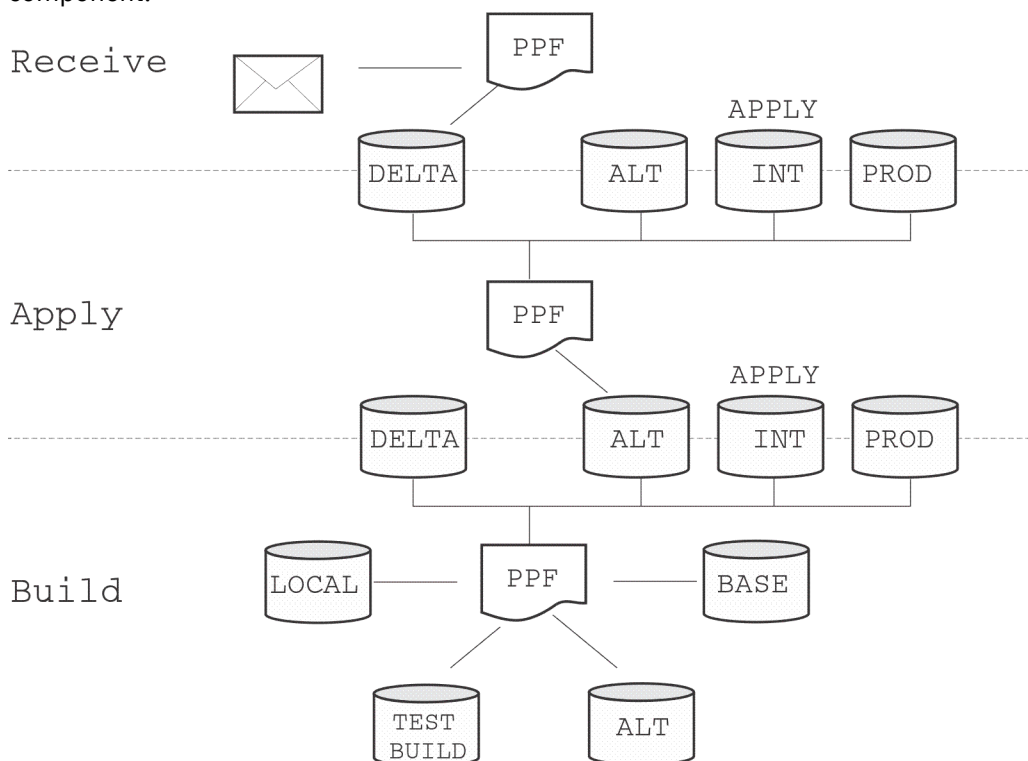


Figure 1. Disks used when installing service

The service and inventory disks are described in 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 E, “Testing service for preinstalled components and features,”](#) on page 75. For the base components of z/VM (CMS, CP, and others), [Appendix D, “Testing service for the base components of z/VM,”](#) on page 69 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 the 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 CMS

nucleus, VM Callable Services Library (VMLIB), Shared File System (SFS) modules, and REXX/VM are all dependent on each other functionally.

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 preventive service, the SERVICE command will install the service for all of 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 might introduce other dependencies between components not shown in [“Component dependencies and service order”](#) on page 5. 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.

Chapter 2. Ordering and retrieving service

You can order z/VM service through the IBM Shopz and IBM ServiceLink tools on the web.

For z/VM 7.4, a change log is provided for all of the linear service that is available for each component since the release of the product. The current change logs are available at:

[z/VM Change logs \(https://www.vm.ibm.com/changelog/\)](https://www.vm.ibm.com/changelog/)

When you are reviewing the available service for a component in the change log, you will want to know the current service level for that component. Use the SERVICE command with the STATUS operand to determine the current 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 displays 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 MAINTvrm user ID as filemode D.

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

```
service compname status
```

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

Example:

```
service cp status
```

```
ENTER
```

```
VMFSRV2195I SERVICE CP STATUS
VFSRV2760I SERVICE processing started
VMFSRV1225I CP (7VMCPR40%CPSPFS) status:
VMFSRV1225I Service Level LVL-0001
VMFSRV1225I Production Level LVL-0001
VMFSRV2760I SERVICE processing completed successfully
```

In this example, LVL-0001 represents the level of service that is currently installed for the CP component, which is helpful when viewing the change log for CP.

For more information, see the [SERVICE EXEC](#) in the [z/VM: VMSES/E Introduction and Reference](#).

You also can use the CP QUERY VARIABLE command to gather component service level information. To do so, the QUERY VARIABLE command output can be filtered to display ZVM.SERVICE.xxxxxxx information only, as done with this sample CMS Pipelines command:

```
pipe cp query variable | Locate W2 /ZVM.SERVICE./ | console
```

For more information about:

- The ZVM.SERVICE.xxxxxxx variables, see [Appendix A, “z/VM service environment variables,”](#) on page 59
- This CP command, see [QUERY VARIABLE](#) in the [z/VM: CP Commands and Utilities Reference](#)
- Using pipes, see the [z/VM: CMS Pipelines User's Guide and Reference](#).

Retrieving service

Service is ordered and delivered electronically. After ordering service, you will be notified by IBM through email when the service is ready to be retrieved.

For z/VM service ordered through IBM Shopz or IBM ServiceLink, the GETSHOPZ utility can be used to retrieve your service. GETSHOPZ supports the following types of orders:

Ordering and retrieving service

- z/VM PTF/APAR orders
- z/VM RSU orders
- z/VM ESO orders
- z/VM-related licensed program orders.

After retrieving the service, follow the instructions that were sent to you electronically with your order to apply it.

The MAINT_{vr}m 500 minidisk can be used to store the service files you retrieve. After the files are on the designated CMS minidisk or SFS directory, continue with the instructions provided with your order. The service envelopes need to be installed on z/VM.

Using GETSHOPZ to obtain service

For z/VM service ordered through IBM Shopz or IBM ServiceLink, the GETSHOPZ utility simplifies the download process by reducing the steps needed to retrieve service. When your z/VM system can connect to the Shopz or ServiceLink download site directly, the direct-to-host mode of GETSHOPZ speeds up the process by eliminating the upload from your workstation to z/VM, verifying that the package came from IBM, and detersing the SERVLINK files. When your z/VM system *cannot* connect to the IBM download site, you can use the workstation-upload mode of GETSHOPZ to upload the data from your workstation. Do not attempt to decompress the files. To ensure that integrity of the data is verified, use GETSHOPZ to upload the files as they were received from IBM.

After configuring your system to use this utility, you will use a web browser to interact with GETSHOPZ, which runs a built-in web interface during the data transfer. For some sample screens and more information on using the web interface to transfer a Shopz or ServiceLink order to a z/VM system, see [Navigating the GetShopz Web Interface \(https://www.vm.ibm.com/service/shopzinf.pdf\)](https://www.vm.ibm.com/service/shopzinf.pdf).

Configuring your system to use GETSHOPZ

The GETSHOPZ utility does not require any special privileges or authorizations to run. However, it is recommended that GETSHOPZ be run while logged on to the MAINTvrm user ID, because the service you obtain will, in general, need to be installed using this user ID. Also, the MAINTvrm 500 disk should be large enough to contain the acquired SERVLINK files for use with the SERVICE command. Make sure you access the MAINTvrm 500 disk before running GETSHOPZ. Use the GETSHOPZ DISK option to specify the 500 disk access file mode.

TCPIP DATA

GETSHOPZ requires a TCPIP DATA file that points to a DNS server that can resolve the host name in the target URL. If you don't already have a properly-customized file on your TCP/IP system disks, consider placing such a file on the MAINT 193 minidisk, where the GETSHOPZ utility is installed.

z/VM TLS Support

The direct-to-host download requires that the root certificate authority (CA) for the IBM SW Order Download Server (<https://deliverycb-bld.dhe.ibm.com/>) is installed in the z/VM TLS certificate database for your TCP/IP stack.

You can use the following pipeline to verify that your DNS lookup works, that your CMS user can connect to the IBM download site, and that the TLS certificate has been installed:

```
pipe i:fanin|tcpclient deliverycb-bld.dhe.ibm.com 443 secure getsecinfo|take|cons|elastic|i:
Host deliverycb-bld.dhe.ibm.com validated TLSv1.2 rsa-aes128-sha 128
time 117.626 ms
Ready;
```

When this test fails, it indicates that GETSHOPZ would not be able to use direct transfer from the IBM download site. The error message might indicate that a TCPIP DATA file is missing. When you receive one of the following messages, your CMS user might not be allowed to connect directly to the IBM download site, but is required to use a proxy service.

```
* FPLTCR1142E Unable to resolve deliverycb-bld.dhe.ibm.com
* FPLTCQ1015E ERRNO 60: ETIMEDOUT
```

Note: When using a proxy gateway for z/VM to download the service, the DNS server only needs to resolve the URL of the proxy; the proxy gateway itself will have access to an external DNS that can resolve the URL of the IBM download site.

Starting GETSHOPZ

The GETSHOPZ utility is installed on the MAINT 193 minidisk. Link and access this minidisk, or use the VMLINK command to acquire it:

```
vmLink maint 193
```

To start the web interface, issue:

```
getshopz
```

This allows you to navigate the application with a web browser. Additional options might be required, depending on your system configuration. The program displays the URL to which to point your browser. For example:

```
GETSHOPZ v2.2 - 2025-03
```

```
Web Interface:  
http://acme.com:37757/
```

To start a browser session, click on the URL on the 3270 display or copy the URL and paste it into your browser address bar. For IBM Personal Communications, see Settings > Hotspots . . .

To stop the web interface, use PF3.

Workstation connection

Navigation in the GetShopz application is done through a web browser. By default, the GetShopz application uses a random port number for the web interface. The port number is kept in GLOBALV, so future invocations will try to use the same port number (in case you want to keep it in the browser bookmarks). The following options might be necessary, depending on your configuration.

- When firewalls between the z/VM system and your workstation require a specific port to be enabled, use the PORT option to specify the port number. You should also reserve that port in the TCPIP configuration to make sure it is not used by other users.
- When your configuration uses a different user ID for the z/VM TCP/IP virtual machine and you have not specified that in the TCPIP DATA file, use the TCPIP option to specify the user ID.
- The HOSTNAME option identifies the URL the browser connects to the web interface on your z/VM system. By default, the host name is taken from TCPIP DATA or retrieved by reverse lookup in DNS. When using a secure connection, the host name should match the TLS/SSL server certificate.

If you need a secure connection between the browser and your z/VM system, a valid server certificate with associated root CA and intermediate CA must be installed in the z/VM TLS certificate database. Make sure you know the label that is assigned to the server certificate, if it isn't set as the default. If needed, adjust the TLSLABEL setting that is associated with this connection.

Browser compatibility

The GetShopz application works with the following web browsers:

- Macintosh: Google Chrome
- Red Hat Enterprise Linux Workstation:
 - Google Chrome - Version 120.0.6099.216
 - Mozilla Firefox - 115.7.0esr
- Windows 10, Windows 11:
 - Google Chrome - Version 121.0.6167.161 (64-bit)
 - Microsoft Edge - Version 121.0.2277.112 (64-bit)
 - Mozilla Firefox - 122.0.1 (64-bit)

User authentication

When the browser connects to the web interface on your z/VM system, the default is to verify that the IP address of the browser matches that of the TN3270 session where the user is logged on. When your security policy does not allow this, or when technical reasons prevent this type of authentication, specify the TOKEN option to get a URL with a token for authentication:

```
getshopz ( token
GETSHOPZ v2.2 - 2025-03

Web Interface:
http://acme.com:37757/?token=FGHSQt9QIxI_etxzRNKKQg
```

To stop the web interface, use PF3.

The token is different with each invocation of GETSHOPZ. This means it is not possible to keep the URL as a browser bookmark.

Security

To enable TLS/SSL for the connection between your browser and the web interface, use the SECURE option. This will display an https: URL for the browser to connect to.

```
getshopz ( secure
```

When the server certificate is not set as the default in the z/VM SSL certificate database, use the TLSLABEL option to specify the label of the server certificate.

```
getshopz ( tlslabel zvm2048
```

Note: The connection with the IBM download site always uses TLS/SSL, independent of the connection between the workstation and z/VM.

Verifying z/VM internet connectivity

The direct-to-host mode requires that your z/VM system can connect to the IBM download site.

For Shopz transfers

Your network policy might only allow internet connectivity through a proxy gateway or a different TCP/IP stack.

1. Proxy gateway

If you need to use a direct proxy gateway for the z/VM connection to the IBM download site, use the PROXY option to specify the URL of an anonymous proxy gateway. For example:

```
getshopz ( proxy http://lnxrmh01.acme.com:3128/
```

To use an HTTP Tunnel Proxy gateway, specify the PROXYMODE TUNNEL option in addition to the PROXY url option.

```
getshopz ( proxy http://lnxrmh01.acme.com:3128/ proxymode tunnel
```

2. Alternative TCP/IP stack

For installations that use different TCP/IP stacks for internal and external traffic, the TCPIPEXT option can be used to point to the TCP/IP stack that must be used to connect to the IBM download site.

For ServiceLink transfers

To verify internet connectivity, issue this command:

```
PIPE (END \) not ftp ftps://deliverycb-bld.dhe.ibm.com/ | x: if pick w1 == ,530, | spec ,Hurray!, | x: | cons
```

If you have connectivity, you will see:

```
Hurray!
Ready;
```

Otherwise, you might see the following messages:

```
FPLTCR1142E Unable to resolve deliverycb-bld.dhe.ibm.com ...
ERRNO 60 ETIMEDOUT
ERRNO 10008 GskCertValidation
```

Specifying GETSHOPZ options

You can use options to specify the output disk to store the service files, various items related to the network configuration, and settings that cover security and authentication aspects. You can specify options in any order. For example:

```
getshopz run ( disk t secure
```

Options that are specified with the (default) RUN operand are added to the default options kept in GLOBALV. For a summary of the various options, issue HELP GETSHOPZ or see [“GETSHOPZ EXEC” on page 62](#).

Default GETSHOPZ options

You can use the DEFAULTS operand to store a set of options in GLOBALV for the subsequent use of GETSHOPZ. The DEFAULTS operand with no options displays the current defaults: For example:

```
getshopz defaults
Default options are: ""
Ready;
```

Suppose you need to add extra options. You might issue:

```
getshopz defaults ( disk t token
Ready;
```

Then, display the current defaults again:

```
getshopz defaults
Default options are: "DISK T TOKEN"
Ready;
```

Digital signature verification (DSV) of z/VM service packages

Before applying service to the z/VM system, you can use GETSHOPZ to verify the authenticity and integrity of service packages. The support for DSV of z/VM service packages relies on the following:

1. For each order, an additional GIMPAF cover letter is provided. This cover letter, which is unique for each order, contains the secure hash values to verify the integrity of all of the parts delivered for the order. The hash value of the cover letter can be verified with the hash stated in the order confirmation email from Shopz or ServiceLink.
2. The GIMPAF2 cover letter contains the digital signature that is used to verify authenticity of the cover letter itself. The cover letter is produced by enhancements in GIMZIP, the utility used to package the parts for the service order. The cryptographic key and certificate to produce the digital signature are also provided.

With this support, GETSHOPZ verifies the integrity of signed service files using the GIMPAF and GIMPAF2 cover letters in flight, leaving a SERVLINK file on disk only when the service is safe to apply. When the required cover letters for the order are not available or not valid, the signed service files (file names and file types in lower case) are put in quarantine to prevent the service from being applied by accident at some later time. The small GIMPAF and GIMPAF2 cover letters are retained as documentation of the received service. It is not possible to verify the integrity of the SERVLINK files without the cover letters.

Using an FTP application proxy service

Use the GETSHOPZ PROXYUSER *pattern* and PROXYPASS *pattern* options to specify the username and password for authentication with the FTP application proxy service. The FTP application proxy service typically uses the username and password to encode various attributes needed to proxy the FTP session:

- The username and password to authenticate for usage of the proxy service
- The protocol and hostname for the target FTP site
- The username and password of the target FTP site to which to connect.

The way these attributes are encoded in the proxy credentials depends on proxy configuration and policy. The PROXYUSER and PROXYPASS options take a pattern to have GETSHOPZ produce the proxy credentials for your proxy configuration. At runtime, variables in this pattern are substituted with the actual values determined during the GETSHOPZ transaction. The name of these substitution variables is enclosed between two \$ signs. Anything else in the pattern is copied as-is.

The \$user\$ and \$pass\$ strings in the pattern are replaced by the username and password for the FTP transfer, as found in the pasted ServiceLink page.

A string starting with \$? causes GETSHOPZ to prompt the user for a password, using the user response to substitute the variable in the template. The text between the \$? and the terminating \$ is shown as a hint for the user to clarify the expected response.

Note: If the pattern contains special characters, such as the at sign (@), make sure the CP TERMINAL setting does not interfere. Escape the special characters or use the CP TERMINAL command to turn off the line-editing function. In the latter case, make sure the character delete function, CHARDEL, is turned off. To check on this, issue:

```
query terminal
```

If the system displays something like this:

```
LINEND # , LINEDEL ¢ , CHARDEL @ , ESCAPE " , TABCHAR |
:
```

use the TERMINAL command to turn CHARDEL off:

```
terminal chardel off
```

Here is an example of the PROXYUSER and PROXYPASS options:

```
proxyuser $user$@james@$site$ proxypass $pass$@$?proxy_password$
```

Suppose the ServiceLink order confirmation gives a username of S7917q0 and a password of secret for the download. The patterns above will cause the prompt to provide a password (while showing the "proxy password" text). When the user responds with qwerty, the FTP logon is done with the following:

```
username: S7917q0@james@acme.com
password: secret@qwerty
```

So, you might issue the following command, for example:

```
getshopz ( ftpproxy https://proxy_url:3128/ proxyuser $user$@$?proxy_user$@$site$ proxypass
$pass$@$?proxy_pass$
```

GETSHOPZ issues this prompt:

```
Please provide value for "proxy_user"
```

You would enter james, for example. This value is not displayed.

GETSHOPZ then issues this prompt:

```
Please provide value for "proxy_pass"
```

You would enter pasxxxx, for example. This value is not displayed.

Note: GETSHOPZ does not support SOCKS5 proxies.

More information

- [“GETSHOPZ EXEC” on page 62](#)
- [GETSHOPZ Utility web page \(https://www.vm.ibm.com/service/getshopz.html\)](https://www.vm.ibm.com/service/getshopz.html)
- [Navigating the GetShopz Web Interface \(https://www.vm.ibm.com/service/shopzinf.pdf\)](https://www.vm.ibm.com/service/shopzinf.pdf)
- [How to Use GETSHOPZ \(on z/VM How to videos \(https://www.vm.ibm.com/education/howto.html\)\)](https://www.vm.ibm.com/education/howto.html).

Chapter 3. Installing preventive or corrective service

This topic describes how to install preventive or corrective service. You will use the SERVICE command to determine your current preventive service level, and then to receive, apply, and build the preventive or corrective service that you ordered. After 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. [Table 1 on page 17](#) lists the z/VM preinstalled components, features, and products, in the usual service order.

Table 1. Preinstalled components, features, and products, in preferred service order

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
Directory Maintenance Facility	DIRM
RACF® Security Server for z/VM	RACF
Performance Toolkit for z/VM	PERFTK
Device Support Facilities R17 for CMS	ICKDSF

Requirements and considerations

Before you begin the service process, read through the following requirements and considerations:

1. Important news about installing service can be found here:
[VM Service News \(https://www.vm.ibm.com/service/news/index.html\)](https://www.vm.ibm.com/service/news/index.html)
2. Updates to the service instructions can be found in the latest version of this publication on the IBM Documentation web page for z/VM 7.4:
[z/VM 7.4 documentation \(https://www.ibm.com/docs/en/zvm/7.4\)](https://www.ibm.com/docs/en/zvm/7.4)
3. The MAINT_{vrn} user ID, where *vrn* is the version, release, and modification level of the z/VM system, must be used (rather than the MAINT user ID), to service all z/VM and preinstalled products, features, and components. You can use MAINT_{vrn} to run the SERVICE command from any system in a single

Steps to install preventive or corrective service

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 MAINTvrm to run the automated service commands (SERVICE and LOCALMOD, for example) on a member that is running z/VM vrm.

4. The default definition for the MAINTvrm user ID includes all of the privilege classes that are required to run the service commands successfully. You should not remove any privilege classes from this user ID. If you are using an external security manager program, you should make sure MAINTvrm is defined as a privileged user.
5. 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.
6. If the DIRMAINT 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.
7. For more information about the SERVICE and PUT2PROD commands, see [z/VM: VMSES/E Introduction and Reference](#).
8. For more information about using the SERVICE command to service VMSES/E-formatted licensed programs that are not preinstalled as part of z/VM installation, see [Chapter 7, “Setting up VMSES/E licensed products to use the SERVICE command,”](#) on page 35.

Steps to install preventive or corrective service

In the following steps, you will use the SERVICE command to install preventive or corrective service.

Notes:

1. To service z/VM vrm, log on to MAINTvrm. In an SSI cluster, log on to MAINTvrm on any member system that is running z/VM vrm 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 COR service with your RSU, install the RSU before you install the COR.

To install preventive (RSU) or corrective (COR) service, perform these steps:

1. Using your site's normal backup procedures, create a complete backup of your system.
2. Log on as MAINTvrm.

```
logon maintvrm
:
z/VM Vv.r.m yyyy-mm-ddhh:mm

ENTER

:
Ready;
```

3. Access and deterse (if necessary) the RSU or COR service.
 - a. IPL CMS.

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

ENTER

Ready;
```

- b. Access the minidisk containing the service files and, if not done when you retrieved the service, 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.

- c. 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

The file name of the documentation envelope.

```
vmfupdat sysmemo
```

Displays the memos that were in the documentation envelope.

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

```
service all envfn1 envfn2 ...
VMFSRV2760I SERVICE processing started
:
VMFSRV2760I SERVICE processing completed
successfully
Ready;
```

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

envfn1

The file name for the first envelope.

envfn2 ...

The file name for the second envelope, and any additional envelopes.

4. Enter VMFVIEW SERVICE to view the SERVICE message log.
5. 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:	Perform the following:
Return code 4	<p>a. You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received.</p> <p>b. Go to substep “6” on page 20.</p>
Return code 6 and the message:	<p>a. You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received.</p> <p>b. 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 4, “Applying local service and local modifications,” on page 23. For more information, see z/VM: VMSES/E Introduction and Reference.</p> <p>c. After you complete the rework, use the VMFUPDAT SYSLMOD command to flag the local modification as reworked.</p> <p>d. Restart by issuing the SERVICE command as displayed in message VMFSRV2310W.</p> <p>e. Continue from substep “4” on page 19.</p>

```
VMFSUII2760I VMFSUFIN
PROCESSING INCOMPLETE DUE
TO LOCAL MODIFCATIONS
```

If you received:	Perform the following:
A return code greater than 6	<ol style="list-style-type: none"> You can ignore any warning messages listed in the notes below. Take appropriate action for any other warning messages you received. Correct all errors reported in the error messages. Restart by issuing the SERVICE command as displayed in the message VMFSRV2310W. Continue from substep “4” on page 19.

Notes:

- You can ignore message VMFSET2206W for minidisk 5E6.
- You can ignore the following messages and the associated VMF1966W message:

```
DMSLIO201W The following names are undefined: ISPLINK ARIPRDI
DMSLIO201W The following names are undefined: DMSDSCSC
DMSLIO201W The following names are undefined: DMSUSRX1 DMSUSRX2
DMSLIO202W Duplicate identifier messages associated with object IOACMAIN MODULE.
DMSLKD004W Warning messages issued messages associated with objects ILBONBL,
            ILBONTR, ILBOREC, ILBORNT, ILBOSND, ILBOSNT, and ILBOSSN.
DMSLIO994W Restrictive RMODE encountered in CSECT filename
```

- You can ignore the following messages if no CP preventive or corrective service, or local modifications, exist for the subject system:

```
VMFBDN1966W The command, $VMFCPM$, completed with return code 4 when issued
            with the argument(s): BLD $$$TLL$ MAP A
```

```
VMFBDN2253W Service content has not been included in object CLOAD MODULE
```

- You can ignore the following message if the indicated product is not licensed or enabled for use on the subject system:

```
VMFSUI1206W Service was not built for product prodid because
            :BUILD was not set to YES in the VM SYSSUF table
```

- Use the VMFUPDAT SYSMEMO command to review any memos that were received with the service.
- Test the service you just installed. If the service is for a base component of z/VM, see [Appendix D, “Testing service for the base components of z/VM,”](#) on page 69 for information about testing new service. For preinstalled products, see [Appendix E, “Testing service for preinstalled components and features,”](#) on page 75. For all other products, refer to their respective program directories.

Steps for placing service into production

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

Notes:

- Before you run PUT2PROD, if you are running the RACF Security Server:

If you received a message that RACF was serviced, you must follow the instructions to bring down the production RACFVM server and bring up the RACMAINT test server in “Putting service into production” in “RACF Security Server” on page 75. If you do not swap the servers before running PUT2PROD, RACF will not be put into production correctly and you will receive message VMFP2P1219E when running PUT2PROD.

- 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

– TCPMAINT

- To put serviced code for z/VM *vrml* into production, log on to MAINT*vrml*. In an SSI cluster where more than one member is running the same level of software, log on to MAINT*vrml* on each member that is running z/VM *vrml* 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.
- If you are running PUT2PROD in an SSI cluster where there are multiple releases of z/VM, you must make sure that all other release MAINT*vrml* user IDs are logged off of all members of the SSI cluster before running PUT2PROD.
- When you run PUT2PROD, it will place all products and components into production that have pending updates. If you want to put only one product or component into production, include that product or component name on the PUT2PROD command:

```
put2prod compname
```

Perform these steps:

1. Log on as MAINT*vrml*.

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

2. IPL CMS.

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

3. Run PUT2PROD.

```
put2prod
VMFP2P2760I PUT2PROD processing started
:
VMFP2P2760I PUT2PROD processing completed successfully
Ready;
```

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:	Perform the following:
Return code 4	<ol style="list-style-type: none"> a. You can ignore any warning messages listed in the note below. Take appropriate actions for any other warning messages you receive. b. Go to substep “6” on page 22.

If you received:	Perform the following:
A return code greater than 4	<ol style="list-style-type: none">You can ignore any warning messages listed in the note below. Take appropriate actions for any other warning messages you receive.If you received message VMFP2P1219E see “RACF Security Server” on page 75.Correct all errors reported in the error messages.Issue IPL CMS.Issue PUT2PROD.If you receive a nonzero return code, repeat substep “5” on page 21.Go to substep “6” on page 22.

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 4. 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 electronic corrective or preventive delivery media. When a severe problem arises and you cannot wait for a corrective or preventive service envelope, you can get emergency service from IBM. This service can be sent to you 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 topic.

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 corrective or preventive service. 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 G, "Typical local modifications,"](#) on [page 79](#) for instructions on using the GENCPBLS command. If you plan to change which EXECs are placed in the CMS installation segment see [Appendix F, "Local modification example,"](#) on [page 77](#).

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 MAINTvrm user ID on a system that is running z/VM vrm.
2. Receive Local Service or Modifications.

If you have IBM local service, load the service to the MAINTvrm 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.
 - i) Update the local version vector table with the VMFSIM CHKLVL command and the LOGMOD option.

- ii) Determine what MACLIBs need to be rebuilt.
 - iii) 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.
 - i) Determine what MACLIBs need to be rebuilt.
 - ii) 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. See the preliminary list of notes in [“Steps for placing service into production” on page 20](#) for special considerations when running PUT2PROD.

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 must 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_{vrn} 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 7.3.0 and one member running z/VM 7.4.0, you must log on to MAINT730 on the system running z/VM 7.3.0 and run the local modification commands; then you must log on to MAINT740 on the system running z/VM 7.4.0 and run the local modification commands there. After 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_{vrn} user ID.
2. Create and apply the local modification.
 - a. Run LOCALMOD:

```
localmod compname partfn partft
```

compname

Component name associated with the product you are modifying (for example one of the COMPNAME values in [Chapter 3, “Installing preventive or corrective service,”](#) on page 17).

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 compname
```

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 must 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 MAINTvrm 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 7.3.0 and one member running z/VM 7.4.0, you must log on to MAINT730 on the system running z/VM 7.3.0 and run the local modification commands; then you must log on to MAINT740 on the system running z/VM 7.4.0 and run the local modification commands there. After 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 MAINTvrm 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 3, “Installing preventive or corrective service,”](#) on page 17).

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 compname
```

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 corrective or preventive service provided electronically.

Note: In an SSI cluster, you must 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 MAINTvrm 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 7.3.0 and one member running z/VM 7.4.0, you must log on to MAINT730 on the system running z/VM 7.3.0 and run the local modification commands; then you must log on to MAINT740 on the system running z/VM 7.4.0 and run the local modification commands there. After 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 MAINTvrm user ID.
2. Apply the local service.
 - a. Run LOCALMOD:
3. Run SERVICE, for each component processed, to build the local service.

```
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 17](#)).

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

```
service compname build
```

Applying local service

```
put2prod compname
```

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.

Chapter 5. Running standalone builds

Occasionally, you might need to rebuild part of your system, even if you haven't installed any new service. This topic 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 MAINTvrn user ID on the system running z/VM vrm where you want to rebuild the nucleus.
2. Run the SERVICE command to build the nucleus:

```
service compname bldnuc
```

where *compname* is CP, CMS, or GCS.

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

```
put2prod compname
```

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, shut down 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 MAINTvrn user ID on the system running z/VM vrm 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 MAINTvrn user ID on the system running z/VM vrm where you want to resave CMS.
2. Run the PUT2PROD command to resave CMS:

```
put2prod savecms
```

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

Chapter 6. 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:

- a. Retrieve them from backup media.
 - b. Generate them, if source was 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 compname
```

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
```

2. Remove the local modification.

Removing service

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

where:

modid is the 1- to 5-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:

- a. Retrieve them from backup media.
 - b. Generate them, if source was maintained, by using VMFHLASM or VMFEXUPD.
3. Use the SERVICE command to rebuild the parts without the local modification:

```
service compname build
```

4. Put the rebuilt parts into production:

```
put2prod compname
```

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 3, “Installing preventive or corrective service,” on page 17](#) for more information.

```
service compname norec
```

3. Put the restored service into production:

```
put2prod compname
```

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
```

where:

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:

- a. Retrieve them from backup media.
 - b. Generate them, if source was 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 compname
```


Chapter 7. 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 17](#).

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 SYSSUF 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 MAINTvrm.

ENTER

```
logon maintvrm
:
Ready;
```

2. Invoke VMFUPDAT to update the VM SYSSUF file.

```
vmfupdat syssuf
```

*** 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		7VMAVS40	000-0000	000-0000	AVS for z/VM 7.4.0
:INSTALL	YES	:INSPPF	SERVP2P	AVS	
:BUILD	YES	:BLDPPF	SERVP2P	AVS	
:INCLUDE	YES	:P2PPPF	SERVP2P	AVSP2P	
CMS		7VMCMS40	000-0000	000-0000	CMS for z/VM 7.4.0
:INSTALL	YES	:INSPPF	SERVP2P	CMS	
:BUILD	YES	:BLDPPF	SERVP2P	CMS	
:INCLUDE	YES	:P2PPPF	SERVP2P	CMSP2P	

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

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

- Press PF6 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             7VMLEN30 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 PF5 to process. The VM SYSSUF table is updated and you exit VMFUPDAT.

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 install preventive or corrective service” on page 18.](#)

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_{vm} user ID in your cluster (MAINT730 and MAINT740, for example) and perform steps 1 thru 6 on each user ID.

Part 2. Using z/VM Centralized Service Management to service z/VM systems

This part includes information about servicing z/VM systems that use z/VM Centralized Service Management (z/VM CSM).

Chapter 8. An overview of z/VM Centralized Service Management (z/VM CSM)

This chapter introduces the concept of *z/VM Centralized Service Management (z/VM CSM)*. With z/VM CSM, you can manage distinct levels of service for a specific group of traditional z/VM systems, locally or remotely, from one central system.

When using z/VM CSM, one system is designated as the *principal system*. This system uses the Shared File System (SFS) to manage service levels for a set of defined *managed systems*, regardless of their geographic location. The principal system builds service levels using VMSES/E service commands. With z/VM CSM, the service process has been enhanced to keep track of available service levels and to manage whichever files are needed to supply a customer-defined service level to a managed system.

The transport of service levels and communicating with the managed systems that are remote is accomplished using a dedicated FTP server, which is customized for exclusive use for z/VM CSM.

Placing received service into production on a managed system is initiated from the principal system after a given z/VM CSM service level package has been fully received on the managed system. Place into production processing can be done at any time, at your discretion, and does not need to be done at the same time the service is built or transported to a managed system.

The new SERVMGR command uses existing VMSES/E commands to apply service and local modifications, to build serviced content, and to drive the transport of z/VM CSM packaged service to managed systems. SERVMGR also manages the new SFS service directories that are associated with each new level of service, as well as information about the systems that are being managed using z/VM CSM.

Note: Systems without TCP/IP connectivity must be customer-managed, with no z/VM CSM capabilities. For systems with no TCP/IP connectivity or for systems in an SSI, see [Chapter 1, “Servicing your system,”](#) on page 3.

A high-level flow of z/VM CSM

A high-level flow of z/VM Centralized Service Management follows.

1. The z/VM CSM structures are initialized:
 - The SERVMGR command with the INITIALIZE operand is issued from the principal system that will initialize data structures and establish that system as the z/VM CSM principal system.
 - An SFS service level structure on the principal system is defined and contains the base files. This structure will contain all future levels of managed service. New service levels can be added and managed with a separate SERVMGR command.
 - The *z/VM CSM service status table* identifies all available service levels built using z/VM CSM functions. This table is initialized with a service level named BASE (for the base product information). It contains information about each service level, including:
 - The service level name
 - The SFS root directory, which identifies the z/VM release
 - The VM component service levels
 - The security manager status of this service level
 - The service level base for this level
 - The build status
 - The date added
 - The last modified date.

2. A new level of service is built that is comprised of selected IBM service deliverables, local modification updates, or both. The new-level service content is built using the SERVICE command, with relevant updates for this service placed in a newly-created SFS structure to maintain this.
3. Systems to be managed are added to the z/VM CSM management group. The *z/VM CSM system status table* contains information about all of the systems that will be managed using z/VM CSM functions. This table is defined when you add your first system. It contains information about each managed system, including:
 - The system name
 - The current service level
 - The pending new service level (optional)
 - The status of the new service level (optional)
 - The security manager status for this system
 - The type of communication used for this system
 - The TCP/IP IP address or fully-qualified host name (remote systems only)
 - The TCP/IP system-unique file transmission parameters (remote systems only)

Note that when you add a system to a z/VM CSM management group, you must use z/VM CSM commands to apply service to that system. All SERVICE commands, except the SERVICE...STATUS command, are blocked on a managed system.
4. The built service level is packaged and transported to the applicable managed system or systems. The z/VM CSM service package consists of separate SERVLINK files for each z/VM component.
5. Communication between the principal system and each remote managed system is handled by a dedicated FTP server that resides on each managed system.
6. Queries issued against a managed system supply information related to its service processing state, such as pending, received, in production, or error status.

System requirements for z/VM CSM

Before setting up z/VM CSM to manage service from a principal system, review the following requirements and restrictions:

1. z/VM CSM does not run on SSI systems. A system in an SSI does not qualify to be managed by z/VM CSM. Neither the principal system nor any of the managed systems can be a member of an SSI cluster.
2. TCP/IP is required to run on all systems involved - the principal system and any managed systems. A dedicated FTP server instance needs to be configured for z/VM CSM use, for remote systems. A CSMSEVER user ID is defined as part of the z/VM 740 system deliverable for this purpose.

Chapter 9, “Planning for and configuring z/VM CSM,” on page 43 provides specific steps for configuring the CSMSEVER FTP server for z/VM CSM. However, note that the majority of the configuration aspects of the CSMSEVER server, such as configuring the use of an external security manager (ESM) to control access to z/VM resources, or for controls for secure FTP connections, are the same as for a conventional FTP server.

For information about setting up a conventional FTP server, see Chapter 6, “Configuring the FTP Server,” in *z/VM: TCP/IP Planning and Customization*.

3. The VMPSFS file pool is used to support z/VM CSM operations on the principal system. In order to initialize z/VM CSM, on your principal system, you must have at least 4000 4K blocks of free space in Group 1 of the VMPSFS file pool and 520000 4K blocks of free space in Group 2. It is recommended that you add additional space to the file pool to accommodate new service levels as well. Each new service level will need the same amount of space as the level it is based on, plus some amount of space for the new service that will be added to the new level. IBM recommends adding several new minidisks and further recommends that the size of each new minidisk be 10016 cylinders or more.

For SFS management information, see *z/VM: CMS File Pool Planning, Administration, and Operation*.

4. When adding systems to a z/VM CSM management group, the PTF and local service on those systems must match, or be a subset of, the z/VM CSM service level at which they are to be added.

Software requirements for z/VM CSM

■ With z/VM 7.4, there are no specific software requirements related to z/VM CSM.

Chapter 9. Planning for and configuring z/VM CSM

To plan for and configure z/VM CSM, read through the information and complete the steps in the following topics:

- [“Overall system configuration changes” on page 43](#)
- [“SFS configuration changes” on page 43](#)
- [“TCP/IP configuration changes” on page 44](#)
- [“Using nicknames for multiple systems” on page 45](#)

Overall system configuration changes

Before you can use z/VM CSM, you need to complete these system configuration steps:

1. On the selected principal system, define the MAINTCSM user ID as a permanent administrative user for the VMPSFS file pool.
2. On the selected principal system, make sure MAINTCSM has read authority for the MAINT`vr`m 5BC minidisk, where `vr`m is the version and release of z/VM that you will be initializing.
3. On the selected principal system, the MAINTCSM user ID will need to use the FORCE command. The default CP privilege class for this command is class A. You need to add this class to the directory entry for MAINTCSM.

If you are using an external security manager (ESM) to manage privilege classes, you must make sure that MAINTCSM has the proper authorization in the ESM.

For more information about managing CP privilege classes, see [z/VM: CP Planning and Administration](#).

SFS configuration changes

Next, complete these SFS configuration steps:

1. On any system that is to be managed using z/VM CSM support, you must define the CSM`SERVE` user ID as a file pool administrator for the VMPSFS file pool by adding this user ID to the DMSPARMS file on the VM`SERV`P 191 minidisk.
2. The storage space of the VMPSFS file pool must be expanded.

While the VMPSFS file pool comes initialized with z/VM 7.4, you will need to add additional storage space to the file pool. When you add a new disk, it is recommended that you make it a full volume size, such as a full 3390 Model 9.

Planning must be done based on the number of service levels you plan to keep in the z/VM CSM service status table. Make sure you have enough space allocated initially to accommodate growth and avoid frequent increases. It is recommended that when a service level is no longer being used, it is deleted.

For SFS management information, see [z/VM: CMS File Pool Planning, Administration, and Operation](#).

Note: Because z/VM CSM service levels are stored in the VMPSFS file pool, it is important that backups of the user data in the VMPSFS file pool are performed periodically so that the data can be restored, if necessary. Specifically, z/VM CSM system-level data and service-level data are stored in the MAINTCSM and CSM`vr`m file spaces in the VMPSFS file pool. The data in these SFS file spaces should be backed up regularly. For more information about backing up SFS user data, see [Backing Up User Data in z/VM: CMS File Pool Planning, Administration, and Operation](#).

TCP/IP configuration changes

For remote systems that will be managed by z/VM CSM, the following TCP/IP configuration changes must be implemented. These changes are commonly made using the TCPMAINT user ID.

1. Modify the SYSTEM DTCPARMS file (or its equivalent) to add an appropriate definition for the CSMERVE server. For example:

```
.*****
.* z/VM CSM FTP Server
.*
.* Note: When this server is activated for use, ensure a :parms. tag
.* definition is included that cites the server-specific
.* configuration file that is to be used.
.*-----
:nick.CSMERVE
:type.server :class.ftp
:parms.CSMERVE CONFIG *
```

2. Modify the TCP/IP server configuration file (PROFILE TCPIP, or its equivalent) to implement these changes.

- a. Update the OBEY statement to ensure that MAINT740 is authorized to use SMSG functions for the CSMERVE FTP server:

```
OBEY
  OPERATOR TCPMAINT MAINT740 MAINT MPROUTE REXECD SNMPD SNMPQE LDAPSRV
ENDOBEY
```

- b. Make sure a FOREIGNIPCONLIMIT statement exists that allows for a maximum of 200 connections for a given foreign IP address:

```
FOREIGNIPCONLIMIT 200
```

- c. Make sure the CSMERVE server is cited as part of the AUTOLOG statement/stanza:

```
AUTOLOG
:
: CSMERVE 0 ; z/VM CSM FTP Server
: FTPERVE 0 ; FTP Server
:
: ENDAUTOLOG
```

- d. Make sure the CSMERVE server is cited as part of the PORT statement/stanza, with ports 4534 and 4535 reserved for its use:

```
PORT
:
: 4534 TCP CSMERVE NOAUTOLOG ; z/VM CSM FTP Server (Data port)
: 4535 TCP CSMERVE ; z/VM CSM FTP Server (Control port)
:
: ; (End of PORT reservations)
```

3. Copy the IBM-supplied CSMERVE SCONFIG file to the TCPMAINT 198 server configuration disk and rename it as CSMERVE CONFIG. Then, review the contents of the CSMERVE CONFIG file and make any changes that pertain to your installation, such as activating statements associated with secure (encrypted) FTP connections.

Note that the supplied configuration sample includes enabled KEEPALIVE and INACTIVE statements, with respective values of ON and 900 (seconds) specified. To ensure that z/VM CSM commands and data transfers associated with the subject system can be processed as needed, it is advised that these statements be altered *only* in consultation with the IBM Support Center.

For the system that is selected to serve as a z/VM CSM principal system, copy the IBM-supplied FTP SDATA file from the TCPMAINT 592 client code disk to the MAINTCSM 191 disk, and rename it as FTP DATA. Then, modify the DataCtTime statement in this file so it cites a value of 720 (seconds):

```
DataCtTime 720
```

Note that this customized FTP DATA file will need to be relocated to the z/VM CSM root directory for a given release (VMPSFS:CSMvrm) after z/VM CSM has been initialized on the principal system. In addition, this customized file should be used as the basis for any *system_name* FTPDATA file that is created to provide system-unique FTP operational characteristics for a specific managed system.

For detailed information about the files and statements discussed in this topic, see [z/VM: TCP/IP Planning and Customization](#).

Using nicknames for multiple systems

You can use the CMS NAMES command to create nicknames for your managed systems or groups of managed systems. On the MAINTCSM user ID, enter the NAMES command to define your nicknames. When you enter the NAMES command, a panel is opened for you. If you want to set up a nickname for one system, enter the nickname in the Nickname: field and the system name in the List of names: field. If you want to set up a nickname for a group of systems, enter the nickname in the Nickname: field and in the List of names: field, enter the names of the systems, separated by a blank, that you would like to include in the group. There are multiple lines for longer lists, and you can also enter nicknames in the list.

For information about creating and using a CMS NAMES file, see the NAMES command in the [z/VM: CMS Commands and Utilities Reference](#).

Chapter 10. Setting up z/VM CSM

Before z/VM CSM is initialized, network configuration and SFS file pool administration must be set up. For information about these topics, see [z/VM: TCP/IP Planning and Customization](#) and [z/VM: CMS File Pool Planning, Administration, and Operation](#).

The SERVMGR command

The SERVMGR EXEC handles all tasks associated with z/VM Centralized Service Management.

SERVMGR command activity and results are stored in a cumulative message log file called \$CSMCMG \$MSGLOG, which is maintained in the z/VM CSM release-level root directory, VMPSFS:CSMvrm. This log includes overall status results that pertain to all SYSTEM or SRVLVL command functions. For example, the overall transport status (success or failure) resulting from use of the SRVLVL SEND function for a given managed system is logged. When errors arise for a managed system, logged information will include references to the name and location (generally, a managed system-specific SFS directory in the MAINTCSM filespace - VMPSFS:MAINTCSM.system_name) in which more detailed messages have been logged. For successful operations, only high-level information is logged in the SERVMGR log file (the intent of this is to prevent this file from becoming less manageable and informative when working with a large number of managed systems).

Detailed message logs created by the SERVMGR SRVLVL SERVICE command or the SERVMGR SRVLVL LOCALMOD command are also stored in the release-level root directory (VMPSFS:CSMvrm). All z/VM CSM and VMSES log files can be viewed using the VMFVIEW command.

During SERVMGR command processing, the MAINTCSM 191 disk is accessed at file mode Z. This access file mode is maintained after the command completes. Files that are customized for using z/VM CSM (a MAINTCSM NAMES file, for example) or for productivity (a PROFILE XEDIT file, for example) should be maintained on the MAINTCSM 191 disk, so these files are available for reference, when needed.

To accommodate effective management of service content for differing levels of z/VM, the SERVMGR command also maintains the following program work files (which should not be renamed or deleted) in the VMPSFS:MAINTCSM root directory:

- SERVMGR EXEC
- \$CSMUTL\$ EXEC
- VMSESMSC COPY
- VMFUME TEXT
- VMFUMB TEXT

For referencing purposes, Shared File System (SFS) aliases for these files are also created in z/VM CSM root directories (VMPSFS:CSMvrm) for respective z/VM levels. Like their VMPSFS:MAINTCSM root directory BASE files, these file aliases should not be renamed or deleted.

These work files and their aliases are maintained and used by the SERVMGR command to allow credentials that are stored in a KEYVAULT database to be used for interaction with any system that is managed using z/VM CSM. They also ensure that any z/VM CSM command functional enhancements, introduced over time, are handled appropriately (accommodated or disallowed), as appropriate for a given z/VM CSM release level.

For a complete description of the SERVMGR command and all of its options, see the [SERVMGR EXEC](#) in [z/VM: VMSES/E Introduction and Reference](#).

The rest of this section takes you through the steps needed to initialize your z/VM CSM management group and to begin using it to manage service for your member systems.

Initialization and system setup

To initialize z/VM CSM on the principal system, log on to the MAINTCSM user ID and enter:

```
SERVMgr INITIALIZE vrm
```

This command will create the root directory for the release, CSMvrm, in the VMPSFS file pool and then it will add the base release files to the VMPSFS file pool as the first service level, VMPSFS:CSMvrm.BASE. The service level status table will be created with one service level, BASE. The BASE service level does not include the initial service that is shipped with the z/VM product.

Initialize z/VM CSM on your z/VM 740 system by entering:

```
SERVMgr INITIALIZE 740
```

The z/VM CSM message log file, \$CSMCMG \$MSGLOG, is created in the VMPSFS:CSM740 root directory. When you enter a SERVMGR command, this directory is accessed as the A disk for the MAINTCSM user. The 191 disk for MAINTCSM will be accessed as file mode Z.

After the SERVMGR INITIALIZE command has completed, relocate the previously-customized FTP DATA file from the MAINTCSM 191 disk to the VMPSFS:CSM740 root directory.

Adding service levels

After z/VM CSM is initialized, additional service levels must be created. Service levels are created with the SERVMGR SRVLVL ADD command:

```
SERVMgr SRVLVL ADD vrm new_service_level BASEDON existing_service_level security_mgr DESC text
```

The z/VM CSM service status table contains the definition for each service level that is available.

In order to add new z/VM systems to the z/VM CSM management group, you need to create a new service level that contains the initial service that was shipped with the z/VM product. You will use the BASE service level as the BASEDON level in the SERVMGR SRVLVL ADD command. For our example, we will add a new service level with no external security manager (ESM) enabled and we will name it LVL1:

```
SERVMgr SRVLVL ADD 740 LVL1 BASEDON BASE NONESM DESC LVL1 shipped with z/VM product
```

This command will define a new SFS structure, VMPSFS:CSM740.LVL1 and then it will copy all the files from the BASEDON level (VMPSFS:CSM740.BASE) into the new structure.

After the new service level is created, you must install the actual service into the new level using the SERVMGR SRVLVL SERVICE command:

```
SERVMgr SRVLVL SERVICE vrm new_service_level SERVICE_command_options
```

Note: When maintenance is processed using the MAINTCSM user ID, any SERVLINK files that are to be referenced must be stored on a minidisk or SFS directory resource that is accessed at CMS file mode C.

For this example, assume that the service file that was shipped with the z/VM product is named LVL-0001 SERVLINK and that file is located on minidisk MAINT740 500. You must link to the MAINT740 500 disk, access it as file mode C, and then enter the SERVMGR command to add the service to the service level:

```
LINK MAINT740 500 500 RR  
ACCESS 500 C  
SERVMGR SRVLVL SERVICE 740 LVL1 ALL LVL-0001
```

SERVMgr will use the SERVICE command and the SERVICE command options that you specified (ALL LVL-0001) to receive and install the service into the service level you specified (LVL1). You will see all the service messages that you are used to seeing when you run the SERVICE command. The messages generated by the SERVICE command are stored in the \$VMFSRV \$MSGLOG file located in

the VMPSFS:CSM740 directory. You can use the VMFVIEW command to view the message logs as you normally would. To view the z/VM CSM overall message log, enter:

```
VMFVIEW CSM
```

If the SERVMGR SRVLVL SERVICE command completes successfully, the service levels for each serviced component are updated to reflect the service level name.

If the SERVMGR SRVLVL SERVICE command completed with errors, resolve any problems as you would for traditional SERVICE command errors, but within the environment of the z/VM CSM service level in use (here, LVL1).

When the stated problems are resolved, run the SERVMGR SRVLVL SERVICE command again, as follows:

```
SERVGR SRVLVL SERVICE 740 LVL1 (RESTART
```

Note that when a restart of the SERVICE command is needed in the z/VM CSM environment, RESTART is specified as a command *option* and not as a command *operand*.

Now that the first service level (LVL1) is set up and contains the PTFs shipped in LVL-0001, you can add additional levels of service to manage by entering:

```
SERVGR SRVLVL ADD 740 LVL2 BASEDON LVL1 NONESM DESC Service level #2 - 0002
```

After you have created the new service level, you would use the SERVMGR SRVLVL SERVICE command to install the service to that level, as you did for the LVL1 level. In general, you will probably build one service level on top of the previous level, thus ensuring a smooth progression of service for your systems.

Note: If you are using an external security manager (ESM) such as RACF or VMSECURE, you will build service levels that will have that ESM built into the CP nucleus. If you have test systems or other systems that do not require an ESM, you will need to build separate service levels without that ESM in the CP nucleus. Every service level will be defined as with or without an ESM.

Adding managed systems

Notes:

- The principal system must have a TCP/IP network set up before you can add any remote managed systems.
- For a candidate remote system, make sure no PUT2PROD actions are pending. The presence of a *system_ID* \$PRODS file on minidisk PMAINT 41D indicates that there are still PUT2PROD actions to be completed.

Now that you have created at least one service level and installed the initial preventive service into that level, you can use that level to add the new systems that you will manage to the z/VM CSM management group.

Decide which systems you will manage. To add these systems to the z/VM CSM system status table, enter:

```
SERVGR SYSTEM ADD 740 system_name service_level add_options
```

The options specified in *add_options* include security manager designation (ESM or NONESM), HOSTNAME (foreign host name or IP address), and COMMTYPE (LOCAL or FTP). If you choose, you can add the principal system to z/VM CSM as a managed system. It would be added with a communication type of LOCAL. If you do this, you would no longer use the MAINT740 user ID on the principal system to apply service to your principal system. All service would be done through z/VM CSM, just as it would be done for other, remote systems.

Identify which local or remote systems will be managed by z/VM CSM and then log on to the MAINTCSM user ID on the principal system.

To add the principal system as a locally-managed (local) system, enter:

```
SERVGR SYSTEM ADD 740 system_name service_level security_mgr COMMTYPE LOCAL
```

For our example, assume the name of our principal system is SYSTEM1, it has only the first preventive service envelope that was shipped with the z/VM product installed, and it is not running an ESM. In this case, you would enter the following command:

```
SERVGR SYSTEM ADD 740 SYSTEM1 LVL1 NONESM COMMTYPE LOCAL
```

To add a remote system, enter:

```
SERVGR SYSTEM ADD 740 system_name service_level security_mgr HOSTNAME host_name COMMTYPE FTP
```

For this example, assume the name of the remote system is SYSTEM2. It also has only the first preventive service envelope that was shipped with the z/VM product installed and there is no ESM running on this system. The IP address for this system is 123.45.67.87. In this case, you would enter the following command:

```
SERVGR SYSTEM ADD 740 SYSTEM2 LVL1 NONESM HOSTNAME 123.45.67.87 COMMTYPE FTP
```

Our system status table will now contain two entries, one for SYSTEM1 and one for SYSTEM2. Because preventive service PTFs that identify preventive service levels are not real PTFs, these systems will have been added to the z/VM CSM system status table with a pending service level. To change the pending level to the current level, we must now send the service level to the systems and then put it into production.

If PTF or local service content differences are identified that prevent addition of a selected system to the z/VM CSM management group, more information about those differences can be obtained using a sample EXEC called LEVELCHK SAMPEXEC.

LEVELCHK SAMPEXEC is included with the VMSES/E component. This sample EXEC can be used to perform a comparison evaluation of the service content of a given z/VM CSM service level and the analogous information that is created for a z/VM CSM managed system (or a system that is intended as such).

The LEVELCHK SAMPEXEC part, which resides on the MAINT740 5B2 disk, can be used to create an executable LEVELCHK EXEC. For information about using the LEVELCHK sample, see the comments in the LEVELCHK SAMPEXEC file.

Packaging and sending service levels

Before service can be sent to a managed system, it must be put into a package that is suitable for transmission to the managed system. This is done using the SERVGR SRVLVL PACKAGE command:

```
SERVGR SRVLVL PACKAGE vrn service_level
```

To create the package for service level LVL1, enter:

```
SERVGR SRVLVL PACKAGE 740 LVL1
```

This command creates a SERVLINK file for each component that was serviced in the service level. The SERVLINK files contain the updates for the service parts, the updated parts, and the information that is needed to update the service inventory files on the managed systems. These are the files that will be sent to the managed systems.

After the packages are built, they can be sent to the managed system.

Note: When sending service to a managed system (local or remote), the MAINTvrn user ID must be logged off of the managed system. MAINTvrn will be autologged by z/VM CSM and it will run the necessary commands while disconnected. Do not try to log on to MAINTvrn while it is receiving commands from z/VM CSM.

This command would be used to send the service to a managed system:


```
SERVMgr SRVLVL SEND vrn service_level system_name
```

If you have defined a nickname for your system or for a group of systems, you can substitute the nickname for *system_name*.

For our example, to send the packages for service level LVL1 to our remote managed system, SYSTEM2, enter:

```
SERVMgr SRVLVL SEND 740 LVL1 SYSTEM2
```

This command will send the SERVLINK files associated with service level LVL1 to SYSTEM2 using FTP. For the local system, the SERVLINK files are accessed directly from the SFS directories by the MAINT_{vrn} user ID. For FTP, you are prompted to supply the z/VM CSM work user ID (CSMWORK) and its password, for FTP interaction with the remote system. The files are sent to the remote system and then received and processed there.

If there is a failure during the remote processing of the SERVLINK files, relevant error messages are returned to the principal system and the command ends. The error messages will be logged in the \$CSMCMG \$MSGLOG file and possibly other message logs. If there are other logs to review, they will be included in the messages in the \$CSMCMG \$MSGLOG. The system status table is updated with pending service status, which indicates that there was a failure of some sort. To check this status information, enter:

```
SERVMgr SYSTEM QUERY 740 system_name
```

After the errors are corrected, the SEND command can be restarted by including the RESTART option on the SERVMGR SRVLVL SEND command:

```
SERVMgr SRVLVL SEND vrn service_level system_name (RESTART
```

Only the incomplete portions of the original command will be retried.

After the SEND command completes successfully, the system status table is updated to indicate that the pending service level has been successfully received. That service level can now be placed into production.

Putting service into production

In order to put service into production on a managed system, the system status table must reflect that the service was successfully received and stored on the managed system's service disks. When the status in the system status table is acceptable, you would put the service into production on the managed system by entering the following command:

```
SERVMgr SYSTEM PUT2PROD vrn system_name
```

As with the SEND command, you can specify a nickname for a single system or for a group of systems. For our example, the following command will put LVL1 into production on SYSTEM2:

```
SERVMgr SYSTEM PUT2PROD 740 SYSTEM2
```

Note that no service level is included on the PUT2PROD command. The service that was last sent to the target system is the service that is put into production.

On the managed system, MAINT_{vrn} is autologged and instructed to run the PUT2PROD command. After the command completes successfully, the z/VM system inventory files are updated as usual to reflect the current service that is in production. With z/VM CSM, the service level for serviced components is always updated with the name of the z/VM CSM service level that has been installed, put into production, or both. You will notice on the output from the SERVICE...STATUS command that the Service Level and Production Level fields accommodate a 16-character name.

On the principal system, when it receives status that the service was successfully put into production, the current service level in the system status table is updated with the pending service level and the pending service level and pending service level status fields are removed.

z/VM CSM service for external security managers

z/VM CSM supports servicing systems that are running external security managers (ESMs), but because of the local modifications to CP that are required to use an ESM, z/VM CSM must keep track of which service levels and which managed systems are using ESMs. This is done by including ESM information on the SERVMGR ADD commands for service levels and for systems. In our previous examples, we specified NONESM, meaning that there was no ESM involved.

In order to create service that is suitable for a system using an ESM, you would first add a service level and specify ESM as one of your *add_options*. For example, let's suppose we are going to enable RACF as the ESM on some of our z/VM systems. Because we will have some systems that are running RACF and some systems that are not running RACF, we will need two separate service level stacks: one with RACF enabled and one without RACF enabled.

So first, we will add a new service level named LVL2_ESM that is based on service level LVL2. We would add this service level to our z/VM CSM environment by entering the following command:

```
SERVMGR SRVLVL ADD 740 LVL2_ESM BASEDON LVL2 ESM DESC LVL2 enabled for RACF
```

The new service level will be added to the z/VM CSM SVCLVL table with an ESM status of PENDING. This is because we have not yet updated the service level to accept service for the ESM. For RACF, the service level must be enabled for RACF service.

The required enablement command is in the *Memo to Users IBM RACF Security Server for z/VM, function level 740*, which you received with your z/VM 740 order. For other ESMs, refer to their documentation on how to enable an ESM in the z/VM CSM environment. You will not need to enable any future service levels that are based on this enabled level. If you want to create a new service level for use by an ESM system that is based on a non-ESM service level, you will need to enable that new service level.

Now that you have a service level that is enabled for your ESM, you can add a system to manage that is running an ESM. For example, suppose we have a system named SYSTEM3. It is running RACF, has IP address 123.45.67.88, and you have installed preventive service level LVL-0002. We would add this system to our z/VM CSM environment by entering the following command:

```
SERVMGR SYSTEM ADD 740 SYSTEM3 LVL2_ESM ESM HOSTNAME 123.45.67.88 COMMTYPE FTP
```

z/VM CSM will verify that the FTP server on SYSTEM 3 is configured, that the RACF ESM is running, and that the service level on SYSTEM3 is at least a subset of the service found in service level LVL2_ESM. If all the conditions are met, SYSTEM3 is added to your z/VM CSM management group.

Now when servicing systems SYSTEM2 and SYSTEM3, z/VM CSM will allow you to only send a service level that has an ESM status of NONESM to SYSTEM2. You will only be able to send service levels with an ESM status of ESM to SYSTEM3. For future service, you would add new service levels for ESM and NONESM service and then use the SERVMGR SRVLVL SERVICE command to add new service to each type of service level.

Chapter 11. Managing a z/VM CSM environment

Now that your z/VM CSM environment is set up, you will need to manage the service levels and the information about your managed systems. Things that you might need to do to maintain your z/VM CSM environment include the following:

- Query systems that are being managed and their current service levels
- Query existing service levels and their contents
- Validate that information in the z/VM CSM status tables match the service on the managed systems
- Update information about managed systems
- Manage service levels and their content
- Delete systems from your z/VM CSM management group
- Delete obsolete service levels
- Monitor and increase available SFS space in the VMPSFS file pool. For SFS management information, see [z/VM: CMS File Pool Planning, Administration, and Operation](#).

Querying systems and service level information

You can display information about the systems you are managing and the service levels you have defined by using the SERVMGR SYSTEM QUERY and SERVMGR SRVLVL QUERY commands.

Systems

To display summary information about all of the systems that you are managing, issue the following command:

```
SERVMSGR SYSTEM QUERY vrn ALL
```

For example, enter:

```
SERVMSGR SYSTEM QUERY 740 ALL
```

The summary information includes the name of each system and the current and pending service levels, if any, for that system:

```
VMFCMG2195I SERVMGR SYSTEM QUERY 740 ALL
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I -----
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.4.0
VMFCSQ4205I -----
VMFCSQ4202I System Information Details for System: SYSTEM1
VMFCSQ4202I ESM Status: NO
VMFCSQ4202I Current Service Level: <None>
VMFCSQ4202I Pending Service Level: LVL1
VMFCSQ4202I Pending Service Level Status: SUBSET
VMFCSQ4200I
VMFCSQ4202I System Information Details for System: SYSTEM2
VMFCSQ4202I Current Service Level: LVL1
VMFCSQ4202I Pending Service Level: <None>
VMFCSQ4202I Pending Service Level Status: <None>
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

If you specify the DETAILS option, additional information is displayed, such as the communication type, host name, and detailed service level information by component. You can query all systems, a single system, a list of systems, or a nickname that you have defined. For example, enter:

SERVGR SYSTEM QUERY 740 SYSTEM2 (DETAILS)

The following output is displayed:

```

VMFCMG2195I SERVGR SYSTEM QUERY 740 SYSTEM2 (DETAILS
VMFCMG2760I SERVGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I -----
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.4.0
VMFCSQ4205I -----
VMFCSQ4202I System Information Details for System: SYSTEM2
VMFCSQ4202I VRM: 740
VMFCSQ4202I Communication Protocol: FTP
VMFCSQ4202I Host Name: 123.45.67.87
VMFCSQ4202I FTP Port: 4535
VMFCSQ4202I FTP Command Operands: <None>
VMFCSQ4202I FTP Data File: <None>
VMFCSQ4202I ESM Status: NO
VMFCSQ4202I Current Service Level: LVL1
VMFCSQ4202I Pending Service Level: <None>
VMFCSQ4202I Pending Service Level Status: <None>
VMFCSQ4203I Service Status Information for system: SYSTEM2
VMFCSQ4203I
VMFCSQ4203I Component      Current From
VMFCSQ4203I Name:          Service Table:
VMFCSQ4203I -----
VMFCSQ4203I AVSSFS             BASE
VMFCSQ4203I CMSSFS             BASE
VMFCSQ4203I CPSFS             LVL1
VMFCSQ4203I DIRMSFS            LVL1
VMFCSQ4203I DVSFS             BASE
VMFCSQ4203I GCSSFS             BASE
VMFCSQ4203I ICKDSFSFS          BASE
VMFCSQ4203I LESFS             BASE
VMFCSQ4203I PERFTKSFS          BASE
VMFCSQ4203I RACFSFS            BASE
VMFCSQ4203I REXXSFS            BASE
VMFCSQ4203I RSCSSFS            BASE
VMFCSQ4203I TCPIPSFS           LVL1
VMFCSQ4203I TSAFSFS            BASE
VMFCSQ4203I VMSESSFS           BASE
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVGR processing completed successfully (RC=0)
Ready;

```

To display information from the SYSTEM1 \$PRODS file, enter:

SERVGR SYSTEM QUERY 740 SYSTEM1 PRODPENDING

The following output is displayed:

```

VMFCMG2195I SERVGR SYSTEM QUERY 740 SYSTEM1 PRODPENDING
VMFCMG2760I SERVGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I -----
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.4.0
VMFCSQ4205I -----
VMFCSQ4202I System Information Details for System: SYSTEM1
VMFCSQ4202I ESM Status: NO
VMFCSQ4202I Current Service Level: <None>
VMFCSQ4202I Pending Service Level: LVL1
VMFCSQ4202I Pending Service Level Status: SUBSET
VMFCSQ4200I
VMFCSQ4202I System Information Details for System: SYSTEM2
VMFCSQ4202I Current Service Level: LVL1
VMFCSQ4202I Pending Service Level: <None>
VMFCSQ4202I Pending Service Level Status: <None>
VMFCSQ4200I
VMFCSQ4203I SYSTEM1 $PRODS COMPONENT Records:
VMFCSQ4203I 7VMCMS40 CMS COMPONENT SERVP2P
VMFCSQ4203I 7VMCPR40 CP COMPONENT SERVP2P
VMFCSQ4203I 7VMDIR40 DIRM COMPONENT SERVP2P
VMFCSQ4203I 7VMPTK40 PERFTK COMPONENT SERVP2P

```

```

VMFCSQ4203I 7VMTCP40 TCPIP COMPONENT SERVP2P
VMFCMG2760I SERVMGR processing completed successfully (RC=0)

```

Service levels

To display summary information about all of the service levels that you have defined, enter:

```
SERVMGR SRVLVL QUERY urm ALL
```

The summary information includes the name of the service level, the root directory for that level, the date it was created and the service level that it was based on, the date it was last modified and the service level status (TEST or STABLE and locked or not locked):

```

VMFCMG2195I SERVMGR SRVLVL QUERY 740 ALL
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I -----
VMFCSQ4205I CSM SRVLVL Query for z/VM Version 7.4.0
VMFCSQ4205I -----
VMFCSQ4200I Service Level: BASE
VMFCSQ4200I Root Directory: VMPSFS:CSM740.
VMFCSQ4200I Service Level added on: 09/12/24
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 09/12/24
VMFCSQ4202I ESM Status: NO
VMFCSQ4200I Service Level build state: STABLE
VMFCSQ4200I Service Level update lock: Yes
VMFCSQ4200I Description:
VMFCSQ4200I Service level: BASE      Created by: MAINTCSM 09/12/24 16:49:01
VMFCSQ4200I Based on: BASE
VMFCSQ4200I Description: z/VM 7.4.0 Base Service Level
VMFCSQ4200I
VMFCSQ4200I Service Level: LVL1
VMFCSQ4200I Root Directory: VMPSFS:CSM740.
VMFCSQ4200I Service Level added on: 09/12/24
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 09/12/24
VMFCSQ4200I Service Level build state: STABLE
VMFCSQ4200I Service Level update lock: Yes
VMFCSQ4200I Description:
VMFCSQ4200I Service level: LVL1      Created by: MAINTCSM 09/12/24 17:15:25
VMFCSQ4200I Based on: BASE
VMFCSQ4200I Description: LVL1 SHIPPED WITH Z/VM PRODUCT
VMFCSQ4206I Full service level description available in SERVLVL DESCRIPT
file in directory: VMPSFS:CSM740.LVL1
VMFCSQ4200I
VMFCSQ4200I Service level: LVL2
VMFCSQ4200I Root Directory: VMPSFS:CSM740.
VMFCSQ4200I Service Level added on: 09/12/24
VMFCSQ4200I Based on Service Level: LVL1
VMFCSQ4200I Service Level last modified on: 09/12/24
VMFCSQ4200I Service Level build state: TEST
VMFCSQ4200I Service Level update lock: No
VMFCSQ4200I Description:
VMFCSQ4200I Service level: LVL2      Created by: MAINTCSM 09/12/24 11:32:15
VMFCSQ4200I Based on: LVL1
VMFCSQ4200I Description: LVL number 2 - LVL-0002
VMFCSQ4206I Full service level description available in SERVLVL DESCRIPT
file in directory: VMPSFS:CSM740.LVL2
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;

```

If you include the DETAILS option on the command, all of the PTFs that are included in a service level for each component are included in the output.

If you specify the SYSTEMS options, for each service level, the output will list any systems that are currently running that level and any systems where that level is pending.

Managing service levels and their content

As service levels are defined, populated with content, and deployed on managed systems, it is important to make sure the content of each service level is controlled properly. This is needed to keep the service level of relevant managed systems synchronized with that of a defined z/VM CSM service level, and to allow for the creation of successive z/VM CSM service levels.

When a z/VM CSM service level is defined, it is designated as having a TEST build state by default. This state allows for PTF or local service to be applied iteratively and built into that service level, and for iterative transport of the same to a managed system, to accommodate service testing and validation. Because the content of a z/VM CSM service level with a TEST build state can be modified, one might consider this service level to be "unfinished". By intent, the SERVMGR SRVLVL SEND command readily accommodates transport of a service level that is in a TEST build state, as its TEST operand is the command default.

While test verification of a service level is in progress, one can use the SERVMGR SRVLVL SETLOCK command to establish temporary control over content changes, through the use of its ON and OFF operands. When a SERVMGR SRVLVL SETLOCK ON command is used, the SERVMGR SRVLVL SERVICE and SERVMGR SRVLVL LOCALMOD commands cannot be used to effect changes to a z/VM CSM service level. An ensuing SERVMGR SRVLVL SETLOCK OFF command would then allow for continued use of those commands.

After the content for a service level content is deemed appropriate and suitably verified, its content should be permanently "finalized" and designated as having a STABLE build state. When a service level is designated with a STABLE build state, no additional alteration of its service content is possible. Having a service level in a STABLE state ensures that invariant service content is distributed to and installed on all of the managed systems that are to make use of that service level.

A STABLE build state is established by sending a service level to at least one managed system, by using the SERVMGR SRVLVL SEND command with the STABLE operand. A service level also becomes stabilized when it is used as the "base" for a successive, newly-defined service level. This association is made by citing the "base" level with the BASEDON operand of the SERVMGR SRVLVL ADD command, which is used to create a new z/VM CSM service level.

To conserve SFS space, you might want to delete service levels that are no longer being used by any systems that you manage. To delete a service level, use the following command:

```
SERVMGR SRVLVL DELETE vrn service_level
```

Maintaining managed system information

To account for changes to the systems that are part of a z/VM CSM management group, regular maintenance of the information z/VM CSM maintains (on the principal system) about those systems is required. Keeping this system and service level information current will help to keep things running smoothly in a z/VM CSM environment. For example:

- If a system name and its associated IP information changes, use the MODIFY option:

```
SERVMGR SYSTEM MODIFY vrn system_name new_system_name HOSTNAME ip_info
```

- If the ESM status of a system changes, use the MODIFY option to update the ESM information:

```
SERVMGR SYSTEM MODIFY vrn system_name ESM  
SERVMGR SYSTEM MODIFY vrn system_name NONESM
```

- If a system is no longer being managed by z/VM CSM, you can delete it from the z/VM CSM environment with the following command:

```
SERVMGR SYSTEM DELETE vrn system_name
```

If the system being removed is a remote system, commands will be sent to the remote system to update the VMSES tables to allow traditional service commands. Therefore, you should remove a system from z/VM CSM before shutting down the z/VM CSM FTP communications.

- To compare service levels on a managed system against the z/VM CSM service level, use the SYSTEM SERVCHECK option:

```
SERVGR SYSTEM SERVCHECK vrm system_name
```

For a complete description of the SERVGR command and all of its options, see the SERVGR EXEC in *z/VM: VMSES/E Introduction and Reference*.

Special service management tasks

A situation might arise for which you need to temporarily remove a managed system from a z/VM CSM environment and apply service manually. This can be done, but this is not recommended by IBM and should be used with caution. The SERVGR MANAGED command can be used to start and stop the use of z/VM CSM service management functions. The command is entered on a managed system. The SERVGR MANAGED OFF command temporarily removes a managed system from the z/VM CSM environment. After the managed system has been removed, if the service is updated, it will have to be managed manually until you re-synchronize it with the principal system. After you have re-synchronized the service with the principal system, you would use the SERVGR MANAGED ON command to reintroduce the system into the z/VM CSM management group.

Upgrading a z/VM CSM-managed system to a new release of z/VM

In order to upgrade a system whose service is managed using z/VM CSM, you must remove that system from the z/VM CSM management group. To remove the system from a management group, log on to MAINTCSM on the principal system and enter the command:

```
SERVGR SYSTEM DELETE vrm sys_name
```

This will update the VM SYSSUF table on the managed system *vrm sys_name* and it will remove *vrm sys_name* from the appropriate z/VM CSM system and service tables on the principal system. After you have upgraded your system to a new release of z/VM, you can resume the use of z/VM CSM to manage service for that system after you have initialized a z/VM CSM service management group for the new level of z/VM.

Deleting a version/release/modification level

If all of the systems in your z/VM CSM group have been upgraded to a new version/release/modification level (*vrm*) of z/VM, you might want to delete the old *vrm* level from your z/VM CSM group.

To delete a *vrm* level from z/VM CSM on the principal system, log on to the MAINTCSM user ID and enter the following command:

```
SERVGR REMOVE vrm
```

This command deletes the root directory for the CSM*vrm* release in the VMPSFS file pool, which includes deleting the CSM SYSSTAT and CSM SVCSTAT tables. Before the *vrm* level is deleted, the SERVGR message log file, \$CSMCMG \$MSGLOG, is copied from the z/VM CSM root directory for *vrm* (VMPSFS:CSM*vrm*.) to the MAINTCSM 191 disk or to SFS directory VMPSFS:MAINTCSM. and saved as \$CSMCMG \$MSG*vrm*.

Note: If a *vrm* level is deleted from the z/VM CSM group using the SERVGR REMOVE command, to use that *vrm* level again for the z/VM CSM group, you need to use the SERVGR INITIALIZE command to add the *vrm* to the group again.

To delete *vrm* level 720, enter:

```
SERVGR REMOVE 720
```

The MAINTCSM 191 minidisk or VMPSFS:MAINTCSM. SFS directory is accessed at filemode A. The SERVMGR message log for 720, \$CSMCMG \$MSGLOG, is copied to the A-disk with the name \$CSMCMG \$MSG720. The root directory for the 720 release, VMFSFS:CSM720., is deleted.

Reverting from z/VM CSM to traditional service management

To stop using z/VM Centralized Service Management (z/VM CSM) to manage multiple systems and revert back to traditional service management of independent z/VM systems, complete the following steps:

1. For each version/release/modification (*vrn*) level in the z/VM CSM group, perform the following tasks on the MAINTCSM user ID on the principal system:

- a. Delete all managed systems at the *vrn* level. For each managed system at the *vrn* level, enter the following command:

```
SERVGR SYSTEM DELETE vrn system_name
```

Note: To determine all of the managed systems that are at the *vrn* level, enter the following command:

```
SERVGR SYSTEM QUERY vrn ALL
```

- b. After all managed systems at the *vrn* level are deleted from the z/VM CSM group, delete the *vrn* level from z/VM CSM with the command:

```
SERVGR REMOVE vrn
```

2. Manually undo the steps documented in Chapter 9, “Planning for and configuring z/VM CSM,” on page 43, which were originally completed to set up and configure z/VM CSM. To remove storage space from the VMPSFS file pool, see the section called “Removing Space from a File Pool” in [z/VM: CMS File Pool Planning, Administration, and Operation](#).

Appendix A. z/VM service environment variables

To get the most up-to-date list of (and more information about) z/VM service environment variables, go to:

z/VM Service Environment Variable List (<https://www.ibm.com/vm/service/zvmsrvev.html>)

A list of the z/VM service environment variables follows:

- ZVM.SERVICE.APAR.*component ppnnnnn*
- ZVM.SERVICE.COMPONENTS *component_list*
- ZVM.SERVICE.LEVEL.*component ffxx*
- ZVM.SERVICE.VERSION.*component v.r.ff.xx*

where:

component

is a z/VM base component that pertains to a given z/VM service level.

ppnnnnn

is the APAR number that maps to the highest-level PTF applied to a subject component. If no applicable APAR exists for a subject component (no service has yet been applied for it), a placeholder APAR number value of 0000 is supplied for that component.

component_list

is the set of z/VM base components that pertain to a given z/VM service level. This list provides a consuming process with the ability to determine the complete set of components for which service-level information is defined/provided. For example:

```
ZVM.SERVICE.COMPONENTS AVS CMS CP DIRM DV GCS ICKDSF LE PERFTK RACF REXX RSCS TCPIP TSAF VMSES
```

ffxx

is the currently-applied numeric feature pack or fix pack level identifier. If no specific level exists for the component, the value 0000 is supplied.

v.r.ff.xx

is the semantic version identifier for the component. If no specific level exists for the component, the value *v . r . 0 . 0* is supplied.

v

is the z/VM version number.

r

is the z/VM release number.

ff

is the feature pack number.

xx

is the fix pack number.

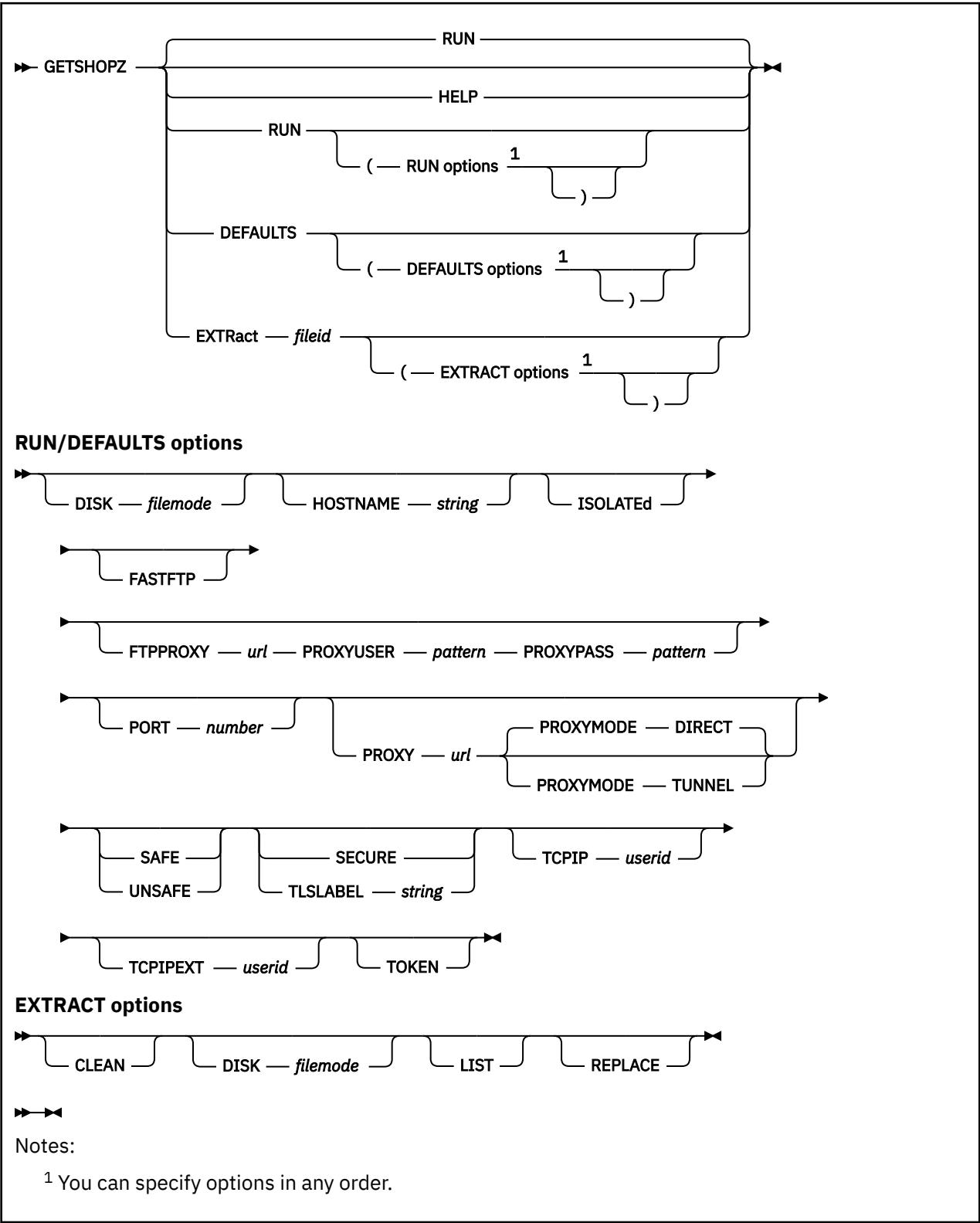
Appendix B. Service-related utilities

This topic includes information about the following:

- [“GETSHOPZ EXEC” on page 62](#)

For information about other service-related commands and EXECs, see [z/VM: VMSES/E Introduction and Reference](#).

GETSHOPZ EXEC



Purpose

Use the GETSHOPZ utility to start the web interface for the transfer of z/VM service packages or z/VM-related licensed program packages from IBM Shopz, IBM ServiceLink, or your workstation.

Operands

RUN

starts the web interface and displays the URL that you will click on or paste into the address bar of a supported web browser.

HELP

displays help information for the GETSHOPZ utility.

DEFAULTS

specifies the default options for subsequent use of the GETSHOPZ utility.

EXTRACT *fileid*

Extracts the contents of service files that were transferred to CMS using a method other than GETSHOPZ. The input file (*fileid*) can optionally be specified with a wildcard to select a set of files to use (S0586978 FILE* A, for example).

Options

DISK *filemode*

specifies the file mode of the disk on which to store the downloaded service files. The default is to use the R/W file mode with the most available space.

HOSTNAME *string*

specifies the host name and domain origin when not using TCPIP DATA or reverse lookup.

ISOLATED

starts a simplified web application that does not require the resources that would otherwise be obtained from external internet sites.

FASTFTP

enables a concurrent FTP session for faster transfer.

FTPPROXY *url*

specifies the URL of an FTP application proxy server to download the service packages.

For more information, see [“Using an FTP application proxy service” on page 15](#).

PROXYUSER *pattern*

specifies the username for authentication with the FTP application proxy service.

PROXYPASS *pattern*

specifies the password for authentication with the FTP application proxy service.

PORT *number*

specifies the number of the reserved port to use for the web interface when using a reserved port.

PROXY *url*

specifies the URL of an anonymous proxy server to download the service packages.

PROXYMODE DIRECT

specifies that the proxy server implements a Direct Proxy.

PROXYMODE TUNNEL

specifies that the proxy server implements an HTTP Tunnel Proxy.

SAFE

enables host name validation for service package download.

UNSAFE

disables host name validation for service package download.

SECURE

specifies that the web interface uses a secure TLS/SSL connection with the default server certificate.

TLSLABEL *string*

specifies that the web interface uses a secure TLS/SSL connection with the named server certificate.

TCPIP *userid*

specifies the user ID of the TCP/IP server to use for the web interface. When this option is not specified, the user ID is taken from the TCPIP DATA file when available, or TCPIP as the default.

TCPIPEXT *userid*

specifies the user ID of the TCP/IP server used to download service packages. When the option is not specified, the user ID from the TCPIP option is used.

TOKEN

enables token-based authentication instead of checking the workstation IP address for authentication of the browser. The TOKEN option is implied when GETSHOPZ can't determine the workstation IP address from the user's TN3270 session.

CLEAN

removes quarantined files when no valid cover letter in the set warrants integrity and authenticity of the files.

LIST

indicates that the input file (S0562114 FILES A, for example) contains the list of files to be processed. The list of files should be in the format *filename filetype filemode* and should be complete, without the use of wildcards. If the line begins with an asterisk (*), it is treated as a comment line and is not processed as a file.

REPLACE

replaces the existing files with the same file name and file type.

Responses

When a service order is transferred to the z/VM system, the order shows that it is signed, and includes the hash value that can be compared to the hash value in the order email. Each file is listed, with the z/VM file name and the original file name. For example:

```
Order S7998815 Signed by IBM Corporation, IBM Code Signing
Order S7998815 Hash Value: 94847FEB6CC85E4BCC1698536D09081B7A62B882
S7998815 GIMPAF B S7998815 Signed GIMPAF.XML
8815DOCS SERVLINK B S7998815 Signed S0001.SHOPZ.S7998815.SHIPDOCS.pax.Z
S7998815 GIMPAF2 B S7998815 Signed GIMPAF2.XML
S7998815 GIMPAF2 B S7998815 Signed GIMPAF2.XML
8815PTFS SERVLINK B S7998815 Signed S0002.SHOPZ.S7998815.SHIPTFSS.pax.Z
```

Usage Notes

1. The GETSHOPZ utility is run from the default MAINTvrm user ID or equivalent.
2. The default TCP/IP server user ID is taken from TCPIP DATA when available.
3. For token-based authentication, a fresh token is generated each time the web interface is started. The displayed URL with the token must be copied and pasted into the browser address field.
4. If the same package is uploaded twice, the older files are kept.
5. Several options apply to direct-to-host download only. These options are ignored when using the workstation-upload mode in the web interface.
6. Use the DEFAULTS operand to specify the default options for the RUN command. The set of options specified replaces previously-set options. The DEFAULTS operand without any options is used to display the current set of options.
7. For GETSHOPZ EXTRACT to verify the contents of the package, all files for the order must be processed in a single invocation of GETSHOPZ. The input files are identified by a wildcard file specification, or with the LIST option. The input files are not removed by GETSHOPZ, so additional disk space is required to also hold the extracted (larger) SERVLINK files.

Use the DISK option to store the SERVLINK files on another file mode. The SERVLINK files are produced only when the GIMPAF2 file was included and the signature has been verified.

There is no need to retain the set of input files after the SERVLINK files have been created. If you do retain the input files and the SERVLINK files and use the EXTRACT function again, GETSHOPZ will

verify the contents of the package and compare them with the SERVLINK file on disk. This could be used as evidence that the SERVLINK file used to apply service was verified.

8. The file mode specified with the DISK option must allow writing and have enough free space to hold the decompressed files to transfer. In practice, the decompressed file may be four times larger than the compressed file that is being transferred.

Examples

1. To start the web interface, issue:

```
getshopz
```

2. To store the downloaded service files on disk T and to specify that the web interface will use a secure TLS/SSL connection with the default server certificate, issue:

```
getshopz ( disk t secure
```

3. To display the current defaults, issue:

```
getshopz defaults
```

4. To add options that store the downloaded service files on disk T and enable token-based authentication, issue:

```
getshopz defaults ( disk t token
```

5. To get a URL with a token for authentication, issue:

```
getshopz ( token
```

6. To specify the URL of an anonymous proxy gateway, issue:

```
getshopz ( proxy http://lnxrmh01.acme.com:3128/
```

7. To bypass host name validation, issue:

```
getshopz ( proxy https://lnxrmh01.acme.com:3128/ unsafe
```

8. To use an HTTP Tunnel Proxy server for direct-to-host transfer, issue:

```
getshopz ( proxy http://lnxrmh01.acme.com:3128/ proxymode tunnel
```

9. To extract files using a wildcard, issue:

```
getshopz extract S0586978 FILE* B
```

10. To extract an order using a list file, issue:

```
getshopz extract S80112219 FILES B (list
```

where the list file (S80112219 FILES B) contains:

```
S8011219 SHIPTFSS B
S8011219 SHIPDOCS B
GIMPAF2 XML B
GIMPAF XSL B
GIMPAF XML B
```

The output will look something like this:

```
----- 13 Feb 2025 10:12:14 -----
Order S8011219 Signed by IBM Corporation, IBM Code Signing
Order S8011219 Hash Value: 289E19DCBB9A6CD55AC192A877ED7BE10E5CFC52
Filename Filetype Fm Order Security Original filename
1219PTFS SERVLINK B S8011219 Signed S8011219 SHIPTFSS B
1219DOCS SERVLINK B S8011219 Signed S8011219 SHIPDOCS B
S8011219 GIMPAF2 B S8011219 Signed GIMPAF2 XML B
S8011219 XSL B S8011219 Signed GIMPAF XSL B
S8011219 GIMPAF B S8011219 Signed GIMPAF XML B
Ready;
```

11. To extract an order using a list file to file mode Z, issue:

```
getshopz extract S0562114 FILES B1 ( list disk z
```

The output will look something like this:

```
----- 14 Feb 2025 11:24:28 -----
Order S0562114 Signed by IBM Corporation, IBM Code Signing
Order S0562114 Hash Value: 717B4658E7CD8D9507EEEC347E693FCF68C69FD9
Filename Filetype Fm Order Security Original filename
S0562114 XSL Z S0562114 Signed S0562114 FILE5 B1
S0562114 GIMPAF Z S0562114 Signed S0562114 FILE4 B1
S0562114 GIMPAF2 Z S0562114 Signed S0562114 FILE3 B1
2114PTFS SERVLINK Z S0562114 Signed S0562114 FILE2 B1
2114DOCS SERVLINK Z S0562114 Signed S0562114 FILE1 B1
Ready;
```

12. To remove quarantined files, use the CLEAN option. For example, suppose you issue:

```
getshopz extract S0562114 FILES-3 B1 ( list
```

And the output looks something like this:

```
----- 15 Feb 2025 12:36:07 -----
Order S0562114 Hash Value: 717B4658E7CD8D9507EEEC347E693FCF68C69FD9
S0562114 XSL Z S0562114 Hash S0562114 FILE5 B1
S0562114 GIMPAF Z S0562114 Hash S0562114 FILE4 B1
ehr3n0eo getshopz Z S0562114 Hash S0562114 FILE2 B1
1kmoss4s getshopz Z S0562114 Hash S0562114 FILE1 B1
Unable to verify authenticity of 2 files; left in quarantine
Ready;
```

To remove the quarantined files, issue:

```
getshopz extract S0562114 FILES-3 B1 ( list clean
```

The output will look something like this:

```
----- 15 Feb 2025 12:40:17 -----
Order S0562114 Hash Value: 717B4658E7CD8D9507EEEC347E693FCF68C69FD9
S0562114 XSL Z S0562114 Hash S0562114 FILE5 B1
S0562114 GIMPAF Z S0562114 Hash S0562114 FILE4 B1
-- file removed -- S0562114 Hash S0562114 FILE2 B1
-- file removed -- S0562114 Hash S0562114 FILE1 B1
Ready;
```


Appendix C. Receiving service for non-VMSES/E envelopes

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

1. Determine whether the service files are in VMSES/E or non-VMSES/E format.

Base your action on the file name and file extension of the service files as they were delivered.

File name and file extension	Description
VLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOCS	VMSES/E-format service documents.
VPTF <i>n..n</i> .bin RPTF <i>n..n</i> .bin <i>ordernum</i> .SHIPTFSS <i>ordernum</i> .SHIPRSUn	VMSES/E-format service files.
RLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOC	Preventive service documents. After they are uploaded, these documents are plain text files. No further action is needed.
NLST <i>n..n</i> .bin <i>ordernum</i> .SHIPDOCN	Non-VMSES/E-format service documents.
NPTF <i>n..n</i> .bin <i>ordernum</i> .SHIPTFSN	Non-VMSES/E-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/E PTF envelopes *nsptfenvfn* SERVLINK.

```
>vmfplcd rst env= nsptfenvfn servlink fm
```

This command resets the logical position to the beginning of the envelope. The GLOBALV will also be updated with the name of the envelope so that the ENV= parameter will not have to be specified in subsequent VMFPLCD commands. *fm* is the file mode of the disk where the envelope exists. The default is the A-disk.

```
vmfplcd scan prodid 010101 (eod
```

Locate the start of service in the envelope for the product to service.

prodid is the product ID for which you want to install service. The *prodid* can be obtained from the COR01 MEDIAMAP file, which was loaded in step 2.

```
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 D. 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 shown in [Table 2 on page 69](#). For other products, these relationships are defined in each product's program directory.

<i>Table 2. Test build and production disks for preinstalled components and features</i>				
Component	Disk owner (user ID)	Test build disks	Disk owner (user ID)	Production disks
VMSES/E	MAINT _{vr} m	490, 493, 49D, 5E6	MAINT	190, 193, 19D, 402, 5E5
REXX/VM	MAINT _{vr} m	490, 493	MAINT	190, 193
LE	vVMLEN _{rm}	49E, 4DD	MAINT	19E, 19D, 402
CMS	MAINT _{vr} m	490, 890, 493, 49D	MAINT	190, 990, 193, 19D, 402
	PMAINT	550	PMAINT	551
CP	MAINT _{vr} m	490, 493, 49D, CF2	MAINT	190, 193, 19D, 402, CF1
	PMAINT	550	PMAINT	551
GCS	MAINT _{vr} m	493, 49D	MAINT	193, 19D, 402
Dump Viewing Facility	MAINT _{vr} m	490, 493, 49D	MAINT	190, 193, 19D, 402
TSAF	MAINT _{vr} m	493, 49D	MAINT	193, 19D, 402
AVS	MAINT _{vr} m	493, 49D	MAINT	193, 19D, 402
TCPIP	vVMTCP _{rm}	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_{vr}m CF2). See [“IPLing with a test level of the CPLOAD MODULE” on page 70](#) for more information.

To test a new CMS nucleus, log on to any user that can link to the MAINTvrn 490 disk. After you log on, enter the command:

```
IPL 490
ENTER
Ready;
```

This will IPL the new, serviced CMS nucleus. You can then try various CMS commands and functions, to make sure the new service is behaving as expected.

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 MAINTvrn 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 *vrnRL1* for systems installed on 3390 DASD, or with label *vrnRES* 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 75.

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

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 4.0
DEVICE NUMBER:    A666      MINIDISK OFFSET:      39  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
```

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 PF10 to load the module and complete the system IPL.

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 4.0
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 the Callable Services Library, Shared File System, and REXX/VM

The CMS nucleus, VM Callable Services Library (VMLIB), Shared File System (SFS) modules, and REXX/VM are all dependent on each other functionally. A minimal test of CMS follows to make sure 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 step.

```
logon userid
```

Log on an SFS server user ID. VMSERV is the recommended user ID, because it is an IBM-supplied server that is used for service installation and maintenance only.

```
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 file (DMSPARMS). The DMSPARMS files for the IBM-supplied servers (VMSERV, VMSERVS, VMSERVER, 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
```

Make sure 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).

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

Testing service for the base components of z/VM

**** 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 MAINTvrm user ID can be used, because it uses the VMSEVP 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 step.	
logon <i>userid</i>	Log on the SFS server user ID used to test the new level of CMS.
stop	Stop the server.
detach 190	Detach the test CMS system disk.
detach 193	Detach the test system tools disk.
define 590 190	Define the production CMS system disk as 190.
define 593 193	Define the production system tools disk as 193.
ipl 190 clear	IPL the old (production) level of CMS.
<i>z/VM Vv.r.m mm/dd/yy hh:mm</i>	
** DO NOT press <u>ENTER</u>! **	
access (noprof	Bypass the execution of the PROFILE EXEC.
access 193 b	Access the production system tools disk.
xedit <i>userid dmsparms</i>	XEDIT the DMSPARMS file for the server.
====> top	Start at the top of the file.
====> set case mixed ignore	Make sure the search is not case significant.
====> locate/nosavesegid	Locate the NOSAVESEGID parameter.
====> change/NOSAVESEGID/SAVESEGID <i>savesegname</i>	Change the NOSAVESEGID parameter to SAVESEGID <i>savesegname</i> . If you are using an IBM-supplied server (VMSERVP, VMSERVS, VMSERVER or VMSERVU), <i>savesegname</i> 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 MAINT_{VTM}

Testing TCP/IP for z/VM service

All new TCP/IP for z/VM service should be tested thoroughly 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 (7VMTCP30, for example, 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 (the FTP or NETSTAT command, for example), only the *tcp_ownerid* 492 and TCPMAINT 592 minidisks need to be accessed (with respect to TCP/IP-specific minidisk requirements).

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 E. 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

Testing new service

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

- If CP and RACF were serviced, as shown by message VMFSRV1233I in the \$VMFSRV \$MSGLOG, you must IPL the system using the CF2 PARM disk and you must specify NOAUTOLOG when you are prompted to enter IPL start options. After the system is IPLed, issue 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 70.
- If only RACF was serviced, you must shut down the production RACFVM server and bring up the new level of code on the backup RACF server, RACMAINT. To do this, log on to an authorized user (such as OPERATOR) and issue the following commands:

```
query sysoper (note the user ID)
set sysoper *
force racmaint
xautolog racfvm
```

Putting service into production

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

1. If the production RACFVM server is running, you should shut it down and bring up the test RACMAINT server:

```
query sysoper (note the user ID)
set sysoper *
force racmaint
xautolog racfvm
```

2. Place the service into production by running PUT2PROD. If necessary, see [“Steps for placing service into production”](#) on page 20.
3. Start up the production RACFVM server.
 - If CP and RACF were both placed into production, 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, issue the following commands to shut down the RACMAINT backup server and bring up the RACFVM production server:

```
force racmaint
xautolog racfvm
set sysoper userid
```

where:

userid is the user ID 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 F. Local modification example

VMSES/E helps you manage your software inventory by keeping track of the products you have installed, which 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 MinRequired constant 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 MinRequired constant (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:

```
VMFREPL2760I VMFREPL processing started
VMFREPL2507I IRMCONS CPYL0002 created on your E-disk from part IRMCONS COPY Q
for use in a VMSES/E environment
VMFREPL2760I VMFREPL processing completed successfully
VMFREPL2760I VMFREPL processing started
VMFREPL2509I The version vector table 7VMCMS30 VVTLCL E will be updated for
the part IRMCONS CPY
VMFREPL2760I 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
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
```

Local Modification Example

```
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_{vm} and run the PUT2PROD command.

Appendix G. 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.
- DMSSBINS 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 DMSSBINS EXEC.

In this example, the C89 EXEC is added to CMSINST. C89 EXEC is in the DMSBLCRX build list in the CMS component.

1. Planning the changes to CMSINST.

If you are adding EXECs to CMSINST, you should ensure that the INSTSEG 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](#).

- 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.

- 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.
 - File your changes by selecting PF5 from the main map panel if they are satisfactory.
- #### 2. Use the LOCALMOD command to update CMSINST LSEG.
- 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 with a **1** to the XEDIT prompt to continue.

b. When you are in the XEDIT session:

```
====> input EXEC C89 EXEC *
====> file
```

```
VMFREP2760I VMFREPL processing started
VMFREP2507I CMSINST SEGL0004 created on your E-disk from
part
CMSINST LSEG Q for use in a VMSES/E environment
VMFREP2760I VMFREPL processing completed successfully
VMFREP2760I VMFREPL processing started
VMFREP2509I The version vector table 7VMCMS30 VVTLC E will
be updated for the part CMSINST SEG
VMFREP2760I 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:

Because the C89 EXEC belongs to the CMS component, the DMSSBINS EXEC must be modified to add a :BLDREQ record to 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 80.

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 with a **1** to the XEDIT prompt to continue.

b. When 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 7VMCMS30 VVTLC E will
be updated for the part DMSSBINS EXC using
the information in AUX file DMSSBINS AUXLC
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 CLOAD 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 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](#).

Typical Local Modifications

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).

```
PRT FILE 0126 SENT FROM MAINT740 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 7VMCPR40 VVTLC 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 CLOAD 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 CLOAD 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 ν rm and run the PUT2PROD command.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785
US

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

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 jurisdictions 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 provide in any way it believes appropriate without incurring any obligation to you.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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 actual people or business enterprises 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.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on [IBM Copyright and trademark information](http://www.ibm.com/legal/copytrade) (<https://www.ibm.com/legal/copytrade>).

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

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 section entitled **IBM Websites** at [IBM Privacy Statement](https://www.ibm.com/privacy) (<https://www.ibm.com/privacy>)
- [Cookies and Similar Technologies](https://www.ibm.com/privacy#Cookies_and_Similar_Technologies) (https://www.ibm.com/privacy#Cookies_and_Similar_Technologies)

Bibliography

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see [z/VM: General Information](#).

Where to Get z/VM Information

The current z/VM product documentation is available in [IBM Documentation - z/VM \(https://www.ibm.com/docs/en/zvm\)](https://www.ibm.com/docs/en/zvm).

z/VM Base Library

Overview

- [z/VM: License Information](#), GI13-4377
- [z/VM: General Information](#), GC24-6286

Installation, Migration, and Service

- [z/VM: Installation Guide](#), GC24-6292
- [z/VM: Migration Guide](#), GC24-6294
- [z/VM: Service Guide](#), GC24-6325
- [z/VM: VMSES/E Introduction and Reference](#), GC24-6336

Planning and Administration

- [z/VM: CMS File Pool Planning, Administration, and Operation](#), SC24-6261
- [z/VM: CMS Planning and Administration](#), SC24-6264
- [z/VM: Connectivity](#), SC24-6267
- [z/VM: CP Planning and Administration](#), SC24-6271
- [z/VM: Getting Started with Linux on IBM Z](#), SC24-6287
- [z/VM: Group Control System](#), SC24-6289
- [z/VM: I/O Configuration](#), SC24-6291
- [z/VM: Running Guest Operating Systems](#), SC24-6321
- [z/VM: Saved Segments Planning and Administration](#), SC24-6322
- [z/VM: Secure Configuration Guide](#), SC24-6323

Customization and Tuning

- [z/VM: CP Exit Customization](#), SC24-6269
- [z/VM: Performance](#), SC24-6301

Operation and Use

- [z/VM: CMS Commands and Utilities Reference](#), SC24-6260
- [z/VM: CMS Primer](#), SC24-6265
- [z/VM: CMS User's Guide](#), SC24-6266
- [z/VM: CP Commands and Utilities Reference](#), SC24-6268

- [z/VM: System Operation](#), SC24-6326
- [z/VM: Virtual Machine Operation](#), SC24-6334
- [z/VM: XEDIT Commands and Macros Reference](#), SC24-6337
- [z/VM: XEDIT User's Guide](#), SC24-6338

Application Programming

- [z/VM: CMS Application Development Guide](#), SC24-6256
- [z/VM: CMS Application Development Guide for Assembler](#), SC24-6257
- [z/VM: CMS Application Multitasking](#), SC24-6258
- [z/VM: CMS Callable Services Reference](#), SC24-6259
- [z/VM: CMS Macros and Functions Reference](#), SC24-6262
- [z/VM: CMS Pipelines User's Guide and Reference](#), SC24-6252
- [z/VM: CP Programming Services](#), SC24-6272
- [z/VM: CPI Communications User's Guide](#), SC24-6273
- [z/VM: ESA/XC Principles of Operation](#), SC24-6285
- [z/VM: Language Environment User's Guide](#), SC24-6293
- [z/VM: OpenExtensions Advanced Application Programming Tools](#), SC24-6295
- [z/VM: OpenExtensions Callable Services Reference](#), SC24-6296
- [z/VM: OpenExtensions Commands Reference](#), SC24-6297
- [z/VM: OpenExtensions POSIX Conformance Document](#), GC24-6298
- [z/VM: OpenExtensions User's Guide](#), SC24-6299
- [z/VM: Program Management Binder for CMS](#), SC24-6304
- [z/VM: Reusable Server Kernel Programmer's Guide and Reference](#), SC24-6313
- [z/VM: REXX/VM Reference](#), SC24-6314
- [z/VM: REXX/VM User's Guide](#), SC24-6315
- [z/VM: Systems Management Application Programming](#), SC24-6327
- [z/VM: z/Architecture Extended Configuration \(z/XC\) Principles of Operation](#), SC27-4940

Diagnosis

- [z/VM: CMS and REXX/VM Messages and Codes](#), GC24-6255
- [z/VM: CP Messages and Codes](#), GC24-6270
- [z/VM: Diagnosis Guide](#), GC24-6280
- [z/VM: Dump Viewing Facility](#), GC24-6284
- [z/VM: Other Components Messages and Codes](#), GC24-6300
- [z/VM: VM Dump Tool](#), GC24-6335

z/VM Facilities and Features

Data Facility Storage Management Subsystem for z/VM

- [z/VM: DFSMS/VM Customization](#), SC24-6274
- [z/VM: DFSMS/VM Diagnosis Guide](#), GC24-6275
- [z/VM: DFSMS/VM Messages and Codes](#), GC24-6276
- [z/VM: DFSMS/VM Planning Guide](#), SC24-6277

- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

Open Systems Adapter

- Open Systems Adapter/Support Facility on the Hardware Management Console (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC14-7580-02.pdf), SC14-7580
- Open Systems Adapter-Express ICC 3215 Support (<https://www.ibm.com/docs/en/zos/2.3.0?topic=osa-icc-3215-support>), SA23-2247
- Open Systems Adapter Integrated Console Controller User's Guide (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC27-9003-02.pdf), SC27-9003
- Open Systems Adapter-Express Customer's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/iaa2z1f0.pdf), SA22-7935

Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

The following publications contain sections that provide information about z/VM Performance Data Pump, which is licensed with Performance Toolkit for z/VM.

- *z/VM: Performance*, SC24-6301. See *z/VM Performance Data Pump*.
- *z/VM: Other Components Messages and Codes*, GC24-6300. See *Data Pump Messages*.

RACF Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320

TCP/IP for z/VM

- [z/VM: TCP/IP Diagnosis Guide](#), GC24-6328
- [z/VM: TCP/IP LDAP Administration Guide](#), SC24-6329
- [z/VM: TCP/IP Messages and Codes](#), GC24-6330
- [z/VM: TCP/IP Planning and Customization](#), SC24-6331
- [z/VM: TCP/IP Programmer's Reference](#), SC24-6332
- [z/VM: TCP/IP User's Guide](#), SC24-6333

Prerequisite Products

Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_3.1.0/pdf/ickug00_v3r1.pdf), GC35-0033

Related Products

XL C[®] ++ for z/VM

- [XL C/C++ for z/VM: Runtime Library Reference](#), SC09-7624
- [XL C/C++ for z/VM: User's Guide](#), SC09-7625

z/OS[®]

IBM Documentation - z/OS (<https://www.ibm.com/docs/en/zos>)

Index

A

- alternate
 - apply disk [7](#)
- APAR descriptions [6](#)
- APPLY disks [6](#)
- applying service
 - overview [4](#)
- audience [xi](#)
- AUX
 - files [4](#)

B

- BASE disks [7](#)
- BUILD disks [7](#)
- builds
 - standalone [29](#)

C

- CMS nucleus
 - rebuilding [29](#)
- CMSINST
 - CMSINST LSEG [79](#)
 - DMSSBINS EXEC [79](#)
 - local modification [79](#)
 - updating [79](#)
- commands
 - SERVMMGR [47](#)
- components
 - order of service [5](#)
 - testing with new service [69](#)
- configuration changes
 - SFS
 - for z/VM CSM [43](#)
 - TCP/IP
 - for z/VM CSM [44](#)
- conventions [xi](#)
- COR service
 - installing [18](#)
- corrective service
 - installing [17](#), [18](#)
 - main tasks [3](#)
 - overview [3](#)
- CP load list
 - updating [81](#)
- CP nucleus
 - rebuilding [29](#)

D

- DELTA disk [6](#)
- Directory Maintenance Facility (DirMaint) [76](#)
- disks
 - used for service [3](#)

- disks used to service z/VM [6](#)
- DMSNGP ASSEMBLE [6](#)

E

- envelopes, non-VMSES/E
 - receiving service for [67](#)
- environment variables
 - z/VM service [59](#)
- ESMs
 - z/VM CSM service for [52](#)
- example
 - local modifications [77](#)
- EXECs
 - SERVMMGR [47](#)
- external security managers
 - z/VM CSM service for [52](#)

F

- files
 - COR VMELEDOC [67](#)
 - COR01 MEDIAMAP [67](#)
 - DMSPARMS [71](#)
 - SERVICE DOCUMENT [67](#)
 - with SERVLINK file type [67](#)

G

- GCS nucleus
 - rebuilding [29](#)
- GETSHOPZ
 - obtaining service with [11](#)
- GETSHOPZ EXEC [62](#)

H

- HCPMDLAT MACRO
 - CP load list [81](#)
 - updating [81](#)
- HELP saved segment
 - example, CMSINST [79](#)

L

- load list
 - updating [81](#)
- local
 - disk [6](#)
 - example, CMSINST [79](#)
 - sample disk [6](#)
- local modification
 - removing [31](#)
- local modifications
 - example [77](#)
- local service

local service (*continued*)

CP load list [81](#)

CPLOAD EXEC [81](#)

example, CMSINST [79](#)

guidelines for update files [24](#)

overview [23](#)

reworking [5](#)

reworking overview [23](#)

LOCALMOD

using [26](#)

LOCALMOD disk [6](#)

LOCALSAM disk [6](#)

LOGO CONFIG [6](#)

M

managed system

putting service into production on [51](#)

managed systems

adding [49](#)

querying [53](#)

merge processing

overview [4](#)

N

nicknames

using for multiple systems [45](#)

non-VMSES/E envelopes

receiving service for [67](#)

O

objects [7](#)

order of service [5](#)

out-of-component requisites [6](#)

P

preface [xi](#)

preventive service

installing [17](#), [18](#)

overview [3](#)

principal system [39](#)

procedures, service

overview [3](#)

production

delta disk [6](#)

production, placing serviced components into

overview [5](#)

procedure [20](#)

PTF

removing a single [31](#)

PTF-numbered parts [6](#)

R

RACF Security Server

testing service [75](#)

receiving service

overview [4](#)

Remote Spooling Communications Subsystem (RSCS) [76](#)

removing service

removing service (*continued*)

overview [31](#)

rework local service [23](#)

RSU

installing [18](#)

S

sample

local modifications [77](#)

saved systems [29](#)

segments [29](#)

sequence of servicing components [5](#)

service

checking level of [9](#)

corrective

installing [18](#)

disks used [6](#)

downloading from ServiceLink [11](#)

downloading from Shopz [11](#)

installing [17](#)

main tasks [3](#)

ordering [9](#)

preventive

installing [18](#)

putting into production

on managed system [51](#)

retrieving [9](#)

supported components, features, and products [17](#)

service levels

querying [55](#)

removing [32](#)

service procedures

overview [3](#)

service, testing

RACF Security Server [75](#)

ServiceLink

obtaining service from [11](#)

SERVMGR EXEC [47](#)

SFS configuration changes

for z/VM CSM [43](#)

Shopz

obtaining service from [11](#)

single PTF

removing [31](#)

standalone builds [29](#)

SYSTEM CONFIG [6](#)

SYSTEM disks [7](#)

system, managed [39](#)

system, principal [39](#)

T

TASK disks [6](#)

tasks

CP load list, updating [82](#)

setting up VMSES/E licensed products to use SERVICE

command [35](#)

TCP/IP configuration changes

for z/VM CSM [44](#)

terminology [xi](#)

testing service

base components [69](#)

testing service (*continued*)

CMS, REXX/VM, SFS, and VMLIB [71](#)

CP [70](#)

Directory Maintenance Facility (DirMaint) [76](#)

preinstalled components [75](#)

Remote Spooling Communications Subsystem (RSCS)
[76](#)

TCP/IP [74](#)

trademarks [86](#)

U

update files [24](#)

utilities

service-related [61](#)

V

variables

z/VM service environment [59](#)

VMFMRDSK EXEC [4](#)

VMFPLCD command [67](#)

VMFREM EXEC [31](#)

VMFSGMAP

example, CMSINST [79](#)

X

xxxMDLAT MACRO

CP load list [81](#)

updating [81](#)

Z

z/VM Centralized Service Management

configuration changes

SFS [43](#)

system [43](#)

TCP/IP [44](#)

configuring [43](#)

environment

managing [53](#)

high-level flow [39](#)

initializing [48](#)

maintaining managed system information [56](#)

managed systems

adding [49](#)

querying [53](#)

overview of [39](#)

planning for [43](#)

service for ESMs [52](#)

service levels

adding [48](#)

managing [56](#)

packaging [50](#)

querying [55](#)

sending [50](#)

servicing z/VM systems with [37](#)

setting up [47](#)

system requirements [40](#)

system setup [48](#)

z/VM CSM

configuration changes

z/VM CSM (*continued*)

configuration changes (*continued*)

SFS [43](#)

system [43](#)

TCP/IP [44](#)

configuring [43](#)

environment

managing [53](#)

high-level flow [39](#)

initializing [48](#)

maintaining managed system information [56](#)

managed systems

adding [49](#)

querying [53](#)

overview of [39](#)

planning for [43](#)

service for ESMs [52](#)

service levels

adding [48](#)

managing [56](#)

packaging [50](#)

querying [55](#)

sending [50](#)

servicing z/VM systems with [37](#)

setting up [47](#)

system requirements [40](#)

system setup [48](#)

z/VM service environment variables [59](#)



Product Number: 5741-A09

Printed in USA

GC24-6325-74

