

z/VM



Getting Started with Linux on System z

version 6 release 2

z/VM



Getting Started with Linux on System z

version 6 release 2

Note:

Before using this information and the product it supports, read the information under “Notices” on page 181.

This edition applies to version 6, release 2, modification 0 of IBM z/VM, (product number 5741-A07) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC24-6194-00.

© **Copyright IBM Corporation 2004, 2011.**

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 used in this document	xi
Where to find more information	xii
Additional Publications	xii
How to send your comments to IBM	xv
If you have a technical problem	xv
Summary of changes	xvii
SC24-6194-01, z/VM Version 6 Release 2	xvii
SC24-6194-00, z/VM Version 6 Release 1	xvii
Chapter 1. About z/VM	1
Example operating system configuration on a single mainframe	3
Overview of a single system image cluster	4
Overview of the Control Program (CP).	5
Central processing units (CPUs)	5
Storage	6
DASD and minidisks	6
Temporary minidisks	6
Virtual disks in storage	6
Virtual readers, punches, and printers	7
The virtual machine console	7
Overview of the CP spool file system	9
The user directory	11
Overview of the Conversational Monitor System (CMS)	14
Minidisks and the CMS access mode	14
CMS files	15
The PROFILE EXEC.	16
The Help system	17
The CMS file editor XEDIT	19
Input mode	22
Overview of changing files.	23
SAVE, FILE, QUIT, and QQUIT	24
Summary of Linux and z/VM similarities.	25
Chapter 2. Planning for Linux virtual servers	27
Overview of z/VM capacity planning	27
Estimating memory and CPU requirements	30
Overview of estimating memory and CPU requirements	30
Guidelines for estimating the amount of DASD you need	33
Estimating DASD for z/VM paging	33
Estimating DASD for the Linux file system	33
Planning your network	34
TCP/IP networking options for Linux	34
Giving Linux virtual servers access to cryptographic hardware for SSL acceleration	36
Planning for user management	37

Planning for single system image clusters and live guest relocation	38
Guidelines for LPAR, memory, and processors in an SSI cluster.	38
DASD guidelines and setup in an SSI cluster.	39
Guideline for saved segments in an SSI cluster	40
Guidelines for planning your network in an SSI cluster	40
Planning your virtual server networks.	43
Cryptographic requirements in an SSI cluster.	43
Tape requirements in an SSI cluster	43
Guideline for user management in an SSI cluster	43
Steps for planning your virtual servers	44
Steps for planning your SSI cluster	44
Steps for obtaining documentation and media	48
 Chapter 3. Changing the system configuration	49
Steps for adding a paging, spooling, temporary disk, or user volume	49
Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG	52
Steps for updating the CP-owned volume list	52
Steps for activating ISFC links between SSI members	56
Steps for updating the user volume list	57
Steps for updating the FEATURES statement	60
Steps for controlling access to devices at startup	62
Steps for defining a virtual switch and defining equivalency IDs for OSA devices	63
Steps for setting addresses for consoles	65
Steps for updating special escape character defaults	67
Steps for checking the syntax of the SYSTEM CONFIG file	68
 Chapter 4. Configuring the Directory Maintenance Facility	71
Steps for enabling DirMaint	71
Steps for adding DirMaint maintenance and changing the passwords for the DirMaint service machines	72
Steps for configuring DirMaint	73
Steps for authorizing users to perform DirMaint tasks.	75
Steps for controlling where DirMaint creates minidisks	75
Steps for putting the configuration into production and starting DirMaint	79
Steps for testing DirMaint	80
Step for modifying the OPERATOR's directory entry	81
How to enter DirMaint commands	81
 Chapter 5. Configuring TCP/IP	83
Setting up the production TCP/IP	83
Steps for automatically starting TCP/IP	83
 Chapter 6. Restarting and checking each z/VM member in the SSI cluster	87
Steps for restarting each z/VM member	87
Steps for checking paging, spooling, and CP-owned volumes.	87
Step for checking the SSI cluster	88
Step for checking the user volume list	89
Steps for checking features	89
Step for checking offline devices	90
Step for checking the virtual switches	90
Step for checking character defaults	91
Steps for checking DirMaint	91
Steps for checking TCP/IP	92
 Chapter 7. Creating your first Linux virtual machine and installing Linux	93
Overview of defining virtual machines for Linux	93

Steps for defining a master virtual machine for Linux	93
Steps for adding CMS to LINMSTR	97
Steps for setting up LINMSTR's disks	99
Installing Linux in a virtual machine	99
Overview of installing Linux in a virtual machine.	99
Example of using FTP to get the Linux boot files	101
Example of punching Linux boot files to the virtual machine reader	103
Example of booting (IPL) the Linux boot files from the virtual machine reader.	103
Chapter 8. Cloning Linux virtual servers	105
Steps for cloning a Linux virtual server.	105
Chapter 9. Setting up basic system automation	107
Starting and stopping virtual machines automatically	107
Steps for automatically starting Linux virtual servers and other virtual machines.	107
Steps for enabling Linux virtual servers to shut down automatically	109
Setting up the programmable operator.	109
Overview of the programmable operator	110
Steps for setting up the routing table	111
Steps for setting up the programmable operator	114
Step for automating Linux virtual consoles	115
Steps for testing your automation.	118
Chapter 10. Performing run-time tasks	119
Overview of console types	119
Real operation tasks	120
Step for monitoring the logical operator console	120
Step for restarting z/VM	122
Step for managing real devices	122
Step for managing users	126
Step for managing the SSI cluster	133
Virtual machine operation tasks	137
Steps for using CP commands at the Linux virtual console	138
Archiving and backing up critical data	141
Overview of archiving z/VM system data	142
Archiving virtual server disks	143
Chapter 11. Monitoring performance and capacity	147
Overview of performance monitoring	147
Monitoring Linux virtual servers with Performance Toolkit for VM	148
Overview of the z/VM scheduler	148
Steps for taking a snapshot of system performance	150
Using the CP Monitor and Performance Toolkit for VM	153
Overview of the CP Monitor and Performance Toolkit for VM	153
Configuring Performance Toolkit for VM	154
Using monitoring to analyze performance and capacity.	163
Steps for using CP commands to improve performance	165
Chapter 12. Servicing z/VM	167
z/VM service concepts	167
Chapter 13. Preparing for live guest relocation	169
What to know before starting relocations	169
Overview of relocation domains	170

Steps for defining relocation domains	171
Steps for placing Linux virtual servers into relocation domains	172
Steps for performing a relocation for a Linux virtual server	174
Steps for troubleshooting guest relocations	175
 Appendix. Example of using FTP to install Linux from the hardware	
management console.	177
Linking the HMC removable media to your z/VM logical partition	177
FTPing to the HMC removable media	178
 Notices	
Trademarks.	183
 Glossary	
 Bibliography	
Where to Get z/VM Information	187
z/VM Base Library	187
Overview	187
Installation, Migration, and Service	187
Planning and Administration.	187
Customization and Tuning	187
Operation and Use	187
Application Programming.	187
Diagnosis	188
z/VM Facilities and Features	188
Data Facility Storage Management Subsystem for VM	188
Directory Maintenance Facility for z/VM	188
Open Systems Adapter/Support Facility	188
Performance Toolkit for VM	189
RACF Security Server for z/VM	189
Remote Spooling Communications Subsystem Networking for z/VM	189
Prerequisite Products	189
Device Support Facilities.	189
Environmental Record Editing and Printing Program.	189
Additional Publications	189
 Index	
 191	

Figures

1. Representation of a mainframe computing system	1
2. Representation of virtual machines.	2
3. Example mainframe configuration	3
4. A four-member z/VM SSI cluster	4
5. Server consolidation example	28
6. Workload distribution patterns	29
7. Relative throughput for patterns A, B, and C.	30
8. Example of a virtual switch	36
9. DASD volumes in an SSI cluster	40
10. Fully connected ISFC topology.	41
11. Logical link	42
12. One logical link composed of multiple CTC devices	42
13. Example of a programmable operator.	110
14. Automating Linux virtual consoles	116
15. General monitoring process	154
16. Initial Performance Data Selection Menu	163

Tables

1. Linux and z/VM similarities	25
2. Task roadmap for configuring TCP/IP and a virtual switch	35
3. Task roadmap for enabling and configuring DirMaint	38
4. Linux virtual server requirements for memory, processors, and devices	46
5. Task roadmap for the production TCP/IP	83
6. Task roadmap for setting up Linux virtual servers	93
7. Linux installation and configuration process	100
8. Decision table for automating Linux virtual consoles	117
9. Decision table for monitoring the logical operator console	121
10. Decision table for managing real devices	123
11. Decision table for managing users	128
12. Decision table for managing the SSI cluster	133
13. Decision table for using CP commands at the Linux virtual console	138
14. Task roadmap for archiving system data	143
15. Decision table for system performance	150
16. How monitor data is collected.	154
17. Tasks for setting up Performance Toolkit for VM	155
18. Resource names for UCOMDIR NAMES file	159
19. Decision table for analyzing performance and capacity	164
20. Decision table for using CP commands to improve performance	166
21. Troubleshooting guest relocations	175

About this document

This document describes how to configure and use z/VM functions and facilities for Linux servers running on the IBM® System z® platform (hereafter referred to as the *mainframe*). The document provides requirements and guidelines to implement during z/VM installation, but primarily assumes you have installed z/VM and are ready to deploy Linux in virtual machines.

Early sections acquaint you with z/VM and take you from the point after z/VM installation to the point where you are ready to install your first Linux server. At that point you must turn to the installation documentation provided by your Linux distributor. Following the deployment of your first Linux server, you can replicate or clone additional servers.

When you finish the early sections, you will have two or more Linux servers running on z/VM with TCP/IP connections to the network. Subsequently, you can turn to vendor-supplied documentation to install applications on Linux. Later sections cover operations, administration, performance, guest relocation, and other day-to-day bare essentials.

Intended audience

This document is designed to help anyone who does system programming, administration, or operation, but has limited knowledge of z/VM and wants to get started deploying Linux servers on z/VM. Before you begin, you must:

- Understand mainframe hardware concepts, such as logical partitions (LPARs) and I/O
- Know and have used the Linux operating system
- Know and have used TCP/IP.

The environment for your z/VM system environment is assumed to include:

- A mainframe with an OSA-Express device
- z/VM version 6 release 2
- Directory Maintenance Facility
- Performance Toolkit for VM
- If you do not have an external file server for the Linux code, you might need an NFS or FTP server.

Conventions and terminology used in this document

This document is primarily a cookbook; that is, it provides instructions about how to accomplish a task or goal. When required, background concepts are provided to help you understand a key z/VM function or facility. Instructions and background concepts are separated but linked together through cross-references, providing an efficient path through the instructional material. At the beginning of each set of instructions, you will see a **Before you begin** section, which explains what you need to know or to do before you perform the task. Cross-references in the **Before you begin** section take you to the necessary background concepts. Thus, if you already know the necessary concepts, you do not need to read those background topics and can simply follow the instructions.

Though the topics in this document are self-sufficient, you might want to explore a function or facility in detail. Some topics end with a list of documents that you can use to understand a function or facility in detail.

In general, new terms (in *italics*) are defined in the context they are introduced.

Sometimes the manual focuses on the virtual machine functions (the virtual hardware) and other times the complete Linux server system (the virtual machine and the Linux operating system as a whole). When focusing on the virtual machine functions only, the term *virtual machine* or *virtual machine for Linux* is used. The term *Linux virtual server* refers to the complete Linux system (virtual machine hardware and the Linux operating system as a whole) running on z/VM.

Commands and statements that you must type are in **bold** while system responses are in normal font.

Example: In the example, you would type “query processors”. The rest of the example is the system response:

```
query processors
PROCESSOR 00 MASTER IFL
PROCESSOR 01 ALTERNATE IFL
PROCESSOR 02 ALTERNATE IFL
PROCESSOR 03 ALTERNATE IFL
PROCESSOR 04 ALTERNATE IFL
:
Ready;
```

Variable information appears in ***bold italics***, which means you must substitute your own values for the variable.

Example: For the command, you would need to supply your own password for the variable ***new_password***.

```
dirm add linmstr like linux pw new_password
```

A vertical ellipsis

```
:
```

indicates system responses that have been removed for clarity.

Where to find more information

For z/VM terms used in this document, see *z/VM: Glossary* or use the z/VM HELP Facility (for more information about glossary terms and the z/VM HELP Facility, see “Glossary” on page 185).

For more information about related publications, see “Bibliography” on page 187.

Additional Publications

For white papers, IBM Redbooks® publications, and other useful information about Linux on the mainframe, visit:

IBM: z/VM resources for Linux on IBM System z (www.vm.ibm.com/linux/)

Another source of Linux information is on the IBM developerWorks® website at:

IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

Links to Other Online Documents

The online version of this document contains links to other online documents. These links are to editions that were current when this document was published. However, due to the nature of some links, 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. Also, a link from this document to another document works only when both documents are in the same directory.

How to send your comments to IBM

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

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

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

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
z/VM V6R2 Getting Started with Linux on System z
SC24-6194-01
- The topic name or page number related to your comment
- The text of your comment

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

IBM or any other organizations will use the personal information that you supply only to contact you about the issues that you submit to IBM.

If you have a technical problem

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

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

Summary of changes

This document contains terminology, maintenance, and editorial changes. Technical changes are indicated by a vertical line to the left of the change. Some product changes might be provided through service and might be available for some prior releases.

SC24-6194-01, z/VM Version 6 Release 2

This edition includes changes to support the general availability of z/VM® V6.2.

Extensive changes have been made throughout the document for single system image cluster and live guest relocation support. Because these changes predominate, there are no change indicators.

SC24-6194-00, z/VM Version 6 Release 1

This edition supports the general availability of z/VM V6.1. Changes made are:

- A step was added to “Steps for copying the current USER DIRECT file.”
- Other minor technical and editorial changes.

Chapter 1. About z/VM

This topic is a z/VM primer and covers general VM concepts, such as editing and finding files, required to complete z/VM system tasks.

When you log onto z/VM, you have the functional equivalent of a real computer and its associated devices at your fingertips. This functional equivalent of a computing system is called a *virtual machine*. Virtual machines are not real, but do work like real systems. Everyone in your entire organization can use z/VM to share the resources of a single computer, while at the same time accessing the system as if each is the only user.

Figure 1 shows a stylized representation of a real mainframe computing system. Each real computing system has one or more central processing units (CPUs), storage (memory), peripheral devices for input and output, such as disks, tapes, printers, and displays, and the operator console. The operating system manages all these resources to do work.

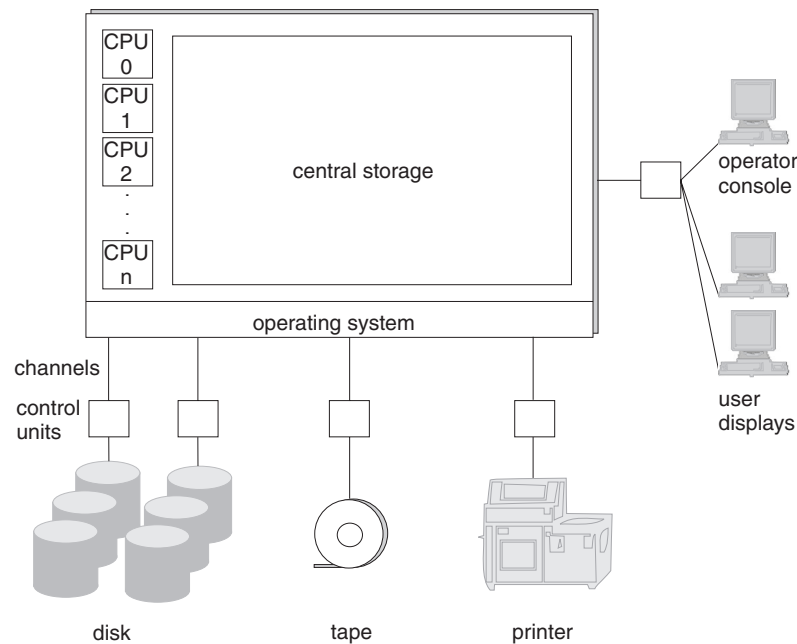


Figure 1. Representation of a mainframe computing system

z/VM virtualizes real computing resources so that each user appears to have a complete mainframe computing system, as shown in Figure 2 on page 2. This means each virtual machine can run its own operating system to manage its virtual resources. It also means you can perform virtual machine tasks as if they were real machine tasks: you can boot (perform an *initial program load* of) an operating system, attach and detach devices, and manage the work of your virtual machine operating system.

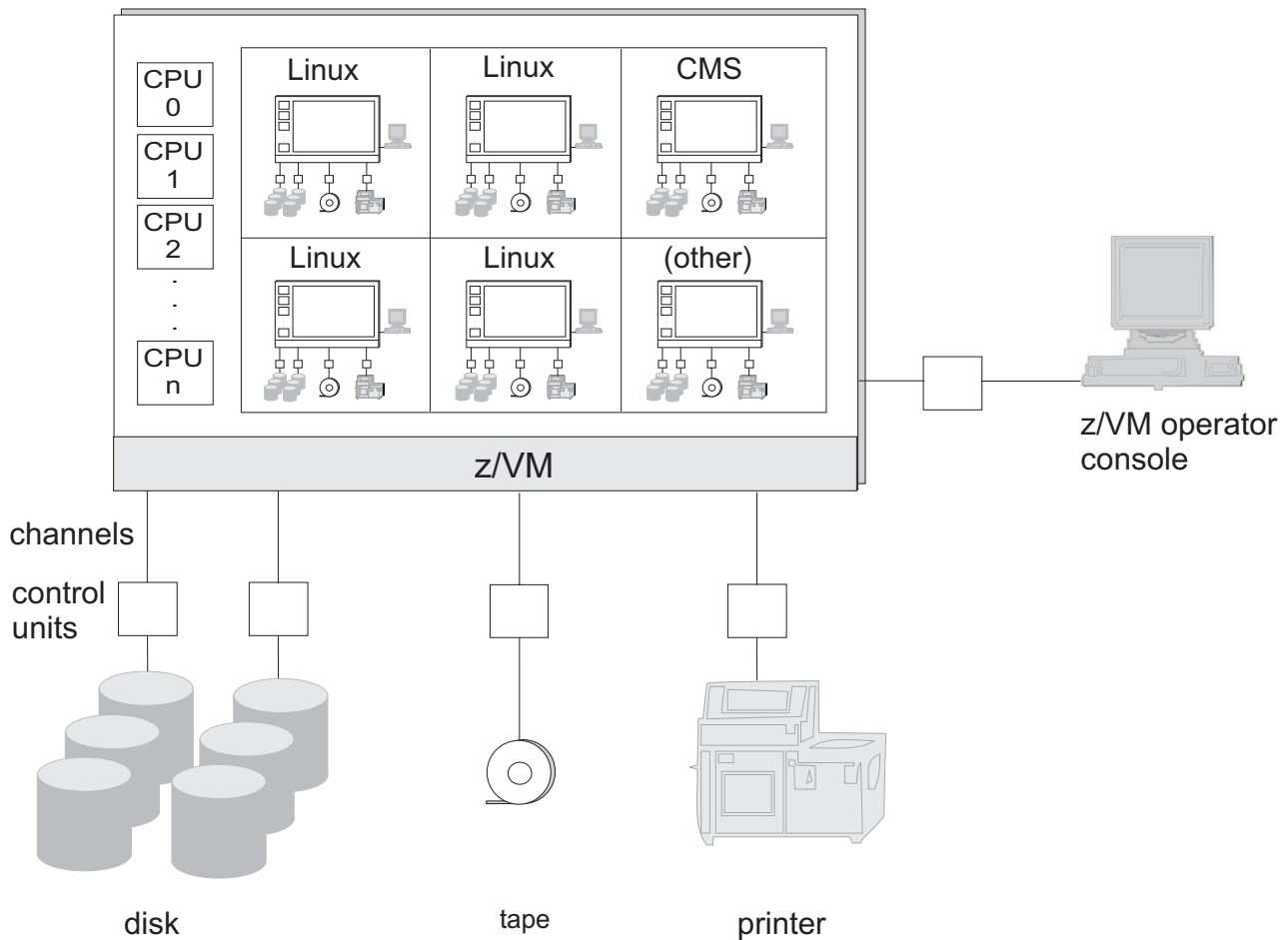


Figure 2. Representation of virtual machines

A virtual machine is directly associated with a z/VM *user ID* or logon identifier. When you log onto z/VM, you have a virtual machine at your disposal and control the virtual machine the way a system operator controls the real hardware.

Some user IDs (virtual machines) are given special privileges to control z/VM and the real machine. For example, the OPERATOR has special privileges allowing control of real machine resources. Another user, usually called MAINT620, has special privileges to change z/VM code and apply z/VM maintenance. Whether or not users have special privileges, they all perform their tasks through a virtual machine. So, as shown in Figure 2, some virtual machines run Linux, while others run other operating systems, such as the Conversational Monitor System (CMS) (more about CMS in a moment).

The Control Program (CP) is the component of z/VM that manages the resources of a single computer so that multiple computing systems (virtual machines) appear to exist. Think of CP as a supervisor (or *hypervisor*) program running in a layer between the hardware and virtual machines. When you are working in the CP environment, you are provided with CPU (central processing unit) functions, input and output devices, and storage. Through CP, each virtual machine can run its own operating system, such as Linux, z/OS®, or z/VM itself.

Operating systems running in virtual machines are often called *guests*. Other terms and phrases you might encounter are:

- *Running first level:* running directly on the hardware, which is what z/VM does.
- *Running second level or running under VM or running on (top of) VM:* running as a guest.

During its time slice, a guest actually runs on the real machine. The hardware microcode handles most guest program instructions and CP takes control only when necessary, which maintains good performance.

The Conversational Monitor System (CMS) is the interactive component of z/VM. CMS is a single-user operating system that runs in a virtual machine. Typically, each directory entry (user definition) has a statement that loads the CMS operating system at logon time (see “The user directory” on page 11). CMS is not only an end-user interactive component, but a home for running system utilities and tasks, such as TCP/IP and system management functions. Systems personnel use the interactive capabilities of CMS to manage z/VM and guests.

Example operating system configuration on a single mainframe

Figure 3 is an example of a mainframe configured for z/VM, Linux guests, and other operating systems.

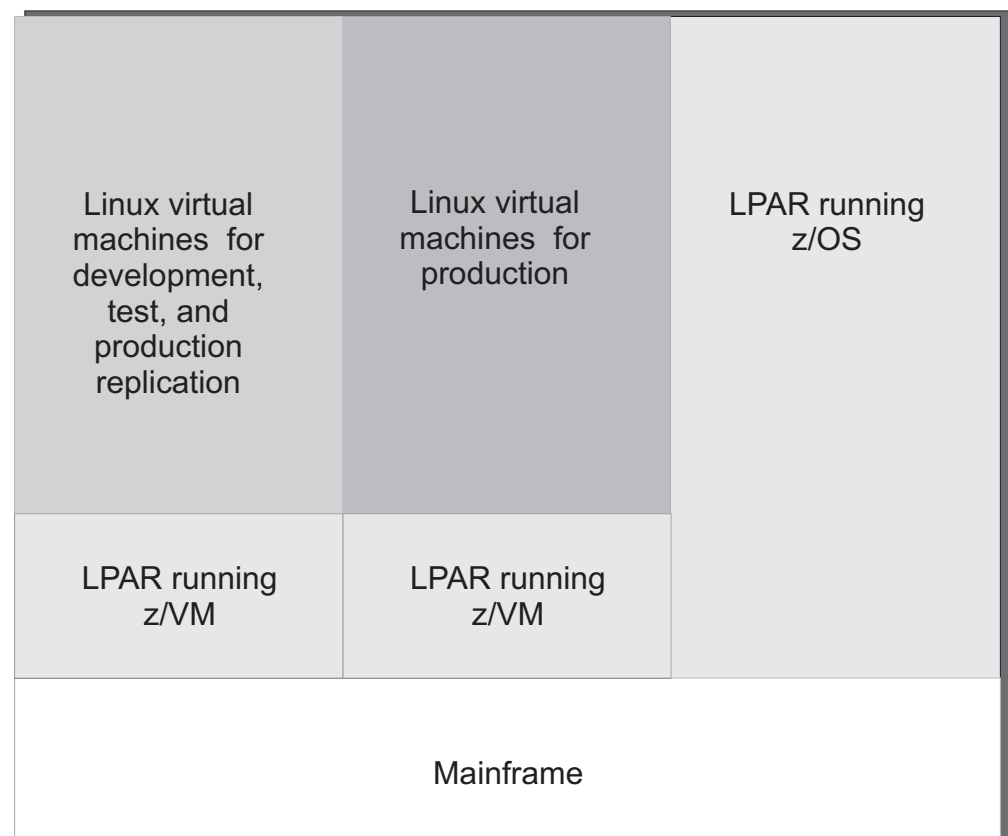


Figure 3. Example mainframe configuration

One logical partition (LPAR) is devoted to Linux production under z/VM. Another LPAR is devoted to Linux application development and test under z/VM; you might also run a replica of your production guests for testing purposes within this LPAR. Finally, another LPAR runs another operating system, such as z/OS.

Overview of a single system image cluster

A z/VM single system image (SSI) cluster is a multisystem environment in which the z/VM systems can be managed as a single resource pool and guests can be moved from one system to another while they are running.

A z/VM SSI cluster consists of up to four z/VM systems in an Inter-System Facility for Communications (ISFC) collection.¹ Each z/VM system is a *member* of the SSI cluster. Figure 4 shows the basic structure of a cluster with four members. The cluster is self-managed by CP using ISFC messages that flow across channel-to-channel devices between the members. All members can access shared DASD volumes, the same Ethernet LAN segments, and the same storage area networks (SANs).

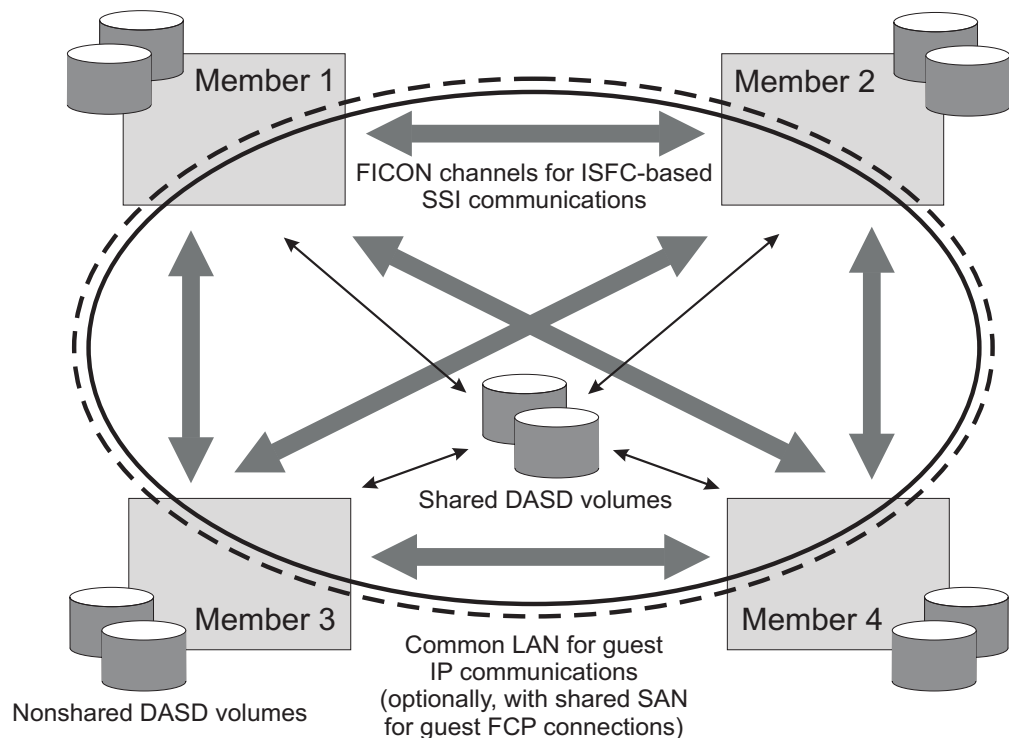


Figure 4. A four-member z/VM SSI cluster

A z/VM SSI cluster provides a virtual server mobility function called *live guest relocation*. A running virtual server (guest virtual machine) can be relocated from one member to another. Relocating virtual servers can be useful for load balancing and for moving workload off a physical server or member system that requires maintenance. After maintenance is applied to a member, guests can be relocated back to that member, thereby allowing you to maintain z/VM and to keep your Linux virtual servers highly available.

Use the VMRELOCATE command to initiate and manage guest relocations. Both the guest to be relocated and the destination environment must meet eligibility requirements, such as the following:

1. ISFC is a function of CP that provides communication services between transaction programs on interconnected z/VM systems. In an SSI cluster, a group of interconnected domains consisting of z/VM systems that use ISFC to communicate with each other is known as an *ISFC collection*.

- The destination member and physical server must provide an architecturally and functionally comparable environment.
- The relocation must meet any policy requirements that have been established. *Relocation domains* can be defined to identify sets of eligible members among which guests can be relocated freely.
- The destination member must have sufficient capacity to accommodate the guest.
- The devices and resources needed by the guest must be shared, so they are also available on the destination member.

When a relocation is initiated, the guest continues to run on the source member until the destination environment is completely prepared. At that point the guest is briefly quiesced on the source member and then resumed on the destination member.

Related information: For a comprehensive introduction to SSI clusters, see “z/VM SSI Environment” in *z/VM: CP Planning and Administration*, SC24-6178.

Overview of the Control Program (CP)

CP acts as a hypervisor layer between the hardware and virtual machines. Each virtual machine appears to have its own CPU, storage (memory), and devices. In reality, these items can be

- Real. For example, you can dedicate a real network interface to a virtual machine for its exclusive use.
- Shared. For example, the CPU is shared through time sharing and real storage is shared as virtual storage. What appears as real storage to a guest is actually virtual storage to CP.
- Simulated. For example, a virtual switch is a simulated LAN networking switch.

CP transparently maps virtual devices and resources to their real counterparts. Topics in this section explain how CP manages computer resources for virtual machines.

Central processing units (CPUs)

A virtual machine can have up to 64 virtual CPUs. If capable of running in multiprocessor mode, your virtual machine operating system dispatches work on its virtual CPUs as if it were running on real hardware. CP handles the dispatching of work on your virtual CPUs to real CPUs.

Guidelines: For CPUs, follow these guidelines:

- Never give a virtual machine more virtual CPUs than there are real CPUs.
- If the Linux program `cpuplugd` is available, define the maximum number of virtual CPUs that a Linux server could use. `cpuplugd` can tailor the virtual CPUs actually used by the Linux server, and thus cause no additional CP overhead. If `cpuplugd` is not available, do not define more virtual CPUs than the Linux server is likely to use.

Usually virtual machines share all CPUs, but a real CPU can be dedicated to a virtual machine, which means that the CPU is reserved for that virtual machine's exclusive use. Unless you have sufficient CPUs available and do good planning, dedicating CPUs can have an impact on the performance of other virtual machines in the system.

Related information: For more information about Linux and CPU usage, see:

IBM: Performance Considerations for Linux Guests (www.vm.ibm.com/perf/tips/linuxper.html)

Storage

Mainframe storage is analogous to memory in a personal computer. CP commands refer to memory as storage, so do not confuse the term “storage” with disk or tape storage.

Each virtual machine has its own virtual storage. CP manages the residency of virtual machine's pages in real storage through paging. Pages that have not been referenced can be moved out of real storage into either expanded storage or onto a paging device. When a virtual machine requires a page no longer in real storage, a page fault occurs and CP brings the missing page back into real storage.

CP has facilities that allow portions of real storage to be shared by many virtual machines. Such portions are called *shared segments*. This sharing economizes on real storage and requires less paging, thereby improving performance. For example, the CMS nucleus is shared in real storage by all virtual machines that loaded CMS by name; that is, every CMS virtual machine maps a 1 MB segment of virtual storage to the same 1 MB of real storage.

DASD and minidisks

DASD, the mainframe term for disk drives, stands for “direct access storage device” and is analogous to a hard disk drive on a personal computer. A single real DASD is called a *volume* or *real volume*. Each volume has a *label* or *volume serial number (volser)* that identifies the volume to z/VM.

Of special importance is the way z/VM shares DASD. CP can logically partition real DASD volumes into *minidisks*, which is analogous to dividing a personal computer hard disk into multiple partitions. A minidisk has its own label, which is distinct from the real DASD label. Each virtual machine can have one or more minidisks and those minidisks are under control of the guest operating system. To the guest, a minidisk appears as an entire DASD volume (though smaller) and the guest runs channel programs as normal to do I/O. Behind the scenes, CP reorients the channel programs: the guest perceives all minidisks as starting at cylinder 0, but the real DASD volume has only one cylinder 0, so CP must modify the cylinder offsets in the channel program to address DASD cylinders owned by the guest.

Temporary minidisks

You can create a temporary minidisk from a special pool of real disks. The disk lasts as long as the virtual machine is logged on. At logoff, the temporary minidisk is deleted and the space returned to the available temporary disk pool.

Virtual disks in storage

Virtual disks in storage are similar to temporary minidisks, except the disks are mapped to storage rather than the cylinders of real disks. Using virtual disks in storage avoids the need for disk I/O. CP manages the virtual disk pages as part of its real memory management.

Virtual readers, punches, and printers

These devices are not associated with real devices, but are implemented through the spool file system. For more information, see “Overview of the CP spool file system” on page 9.

The virtual machine console

The virtual machine console or *virtual console* is the primary interface to the virtual machine. When you log on to a virtual machine from a local terminal or a remote workstation, the virtual console is associated with the terminal session. From the console, you can enter CP commands, such as loading (IPL) an operating system. The virtual console is the device an operating system views as its system or hardware console.

Note: The key assignments for your keyboard might differ from standard key assignments. Some 3270 emulators allow you to remap the key assignments; for example, the Clear function might be assigned to the ESC key on some keyboards and the Pause/Break key on others. Consult your display documentation for key mappings.

As you do work on your console, the lower right corner of the screen displays various status notices. The notices tell you what is happening in the system at the present time. If you forget what these notices mean, you can come back to this section for reference.

CP READ

This notice means that the Control Program (CP) is waiting for you to enter a command.

VM READ

This notice means that a virtual machine operating system, such as CMS, is waiting for you to enter a command.

RUNNING

This notice means the virtual machine is working on something. For CMS, this means CMS is ready for you to enter a command.

MORE ...

This notice means that there is more information than can fit on the current screen. After a pause (which depends on the terminal settings for your virtual machine), the next screen of information is displayed. To view the next screen right away, press the Clear key. To hold this information on the screen, press the Enter key, which changes **MORE...** to **HOLDING**.

HOLDING

This notice means the system is waiting for you to clear the screen before showing you more information. The notice appears when the screen displays **MORE...** and you press the Enter key. The notice can also appear when another user sends you a message. To cancel the hold, press the Clear key.

NOT ACCEPTED

This notice means that the system is working on something and is too busy to accept another command. Wait several seconds and issue your command again.

Related information: For more information about virtual consoles, see “Using a Virtual Machine Operator's Console” in *z/VM: Virtual Machine Operation*, SC24-6241.

CP commands

CP provides you with commands that allow you to view and manipulate your virtual hardware (virtual CPUs, virtual storage, minidisks, and other devices). To issue some CP commands, you need to be in a special privilege class assigned to you in the user directory. Privilege classes are denoted by the letters A through Z, the numbers 1 through 6, or the word "Any." For the tasks explained in this document, the user IDs you use have all the required privilege classes (like superusers in Linux).

Related information: For more information about privilege classes, see "Privilege Classes" in *z/VM: CP Commands and Utilities Reference*, SC24-6175.

Examples of using the CP commands: QUERY displays information about your virtual machine.

1. To display virtual CPUs, class G users can issue the QUERY VIRTUAL CPUS command:

```
query virtual cpus
CPU 00 ID FF0D312E20978000 (BASE) IFL CPUAFF ON
Ready;
```

The response tells you the virtual machine has one base virtual CPU whose address is 00.

2. To display available storage (memory), class G users can issue the QUERY STORAGE command:

```
query virtual storage
STORAGE = 512M
Ready;
```

The response tells you the virtual machine has 512 megabytes of storage.

3. To display information about minidisks, class G users can issue the QUERY VIRTUAL DASD command:

```
query virtual dasd
DASD 0120 3390 PK5001 R/O      250 CYL ON DASD  810B SUBCHANNEL = 000D
DASD 0190 3390 SYGEMC R/O      130 CYL ON DASD  7355 SUBCHANNEL = 0005
DASD 0191 3390 USGE24 R/W       10 CYL ON DASD  7378 SUBCHANNEL = 0000
DASD 019A 3390 PK5001 R/O      400 CYL ON DASD  810B SUBCHANNEL = 0009
DASD 019D 3390 US7E0K R/O      250 CYL ON DASD  801C SUBCHANNEL = 0006
```

The response tells you the virtual machine has 5 minidisks, one of which is read/write (R/W); the others are read only (R/O).

4. To display information about network devices such as the Open Systems Adapter, class G users can issue the QUERY VIRTUAL OSA command:

```
query virtual osa
OSA 700C ON OSA 700C SUBCHANNEL = 0020
700C DEVTYPE OSA VIRTUAL CHPID 70 OSD REAL CHPID 70
700C QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
OSA 700D ON OSA 700D SUBCHANNEL = 0021
700D DEVTYPE OSA VIRTUAL CHPID 70 OSD REAL CHPID 70
700D QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
OSA 700E ON OSA 700E SUBCHANNEL = 0022
700E DEVTYPE OSA VIRTUAL CHPID 70 OSD REAL CHPID 70
700E QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
```

5. To display the size of real storage, class B and E users can issue the QUERY STORAGE command:

```
query storage
STORAGE = 10G CONFIGURED = 70G INC = 1G STANDBY = 240G RESERVED = 0
```

Note: QUERY VIRTUAL *option* displays information about the virtual machine. The keyword “VIRTUAL” is optional for the class G user. For privileged users (those with privilege classes other than G), using QUERY without the keyword “VIRTUAL” displays information about the real machine. For instance, QUERY VIRTUAL STORAGE displays the virtual storage size of the virtual machine while QUERY STORAGE (class B and E) displays the LPAR storage size.

Related information: For more information about CP commands, see:

z/VM: CP Commands and Utilities Reference, SC24-6175

Overview of the CP spool file system

In the early days of computing, input to the computer came from punched cards loaded into a card reader. You used a key punch to record your program on punched cards, then loaded the cards into a card reader, which interpreted your cards and loaded your program into the computer. Output from the program was written to a printer. z/VM preserves this bit of computing history through virtual reader, punch, and printer devices, also called *unit record devices*. Unit record devices provide a handy way to send files from one virtual device to another, to other virtual machines, or to real devices (such as real printers). For instance, you can think of a file being sent from one virtual machine to another as the virtual equivalent of taking a card stack from one computer and loading the stack onto another computer's card reader.

Behind the manipulation of these files is a CP file system called the *spool file system*. CP manages spool files on one or more DASD volumes that act as temporary storage areas. A spool file is a collection of data along with device control instructions for processing on a unit record device. Spooling is the processing of files created by or intended for virtual readers, punches, and printers. Through CP and CMS commands, you can send spool files from one virtual device to another, from your virtual machine to another, and to real devices.

By convention, each virtual machine has a virtual reader at virtual device number 00C, a virtual punch at virtual device number 00D, and a virtual printer at virtual device number 00E. Your virtual reader is like the in-box of an e-mail system, except more than just e-mail can be placed there. Through your virtual punch, you can place a copy of an entire operating system into the system spool, then use the CP IPL command to load and run that operating system in your virtual machine. “Installing Linux in a virtual machine” on page 99 shows you how to use this z/VM facility.

In an SSI cluster, spool volumes (disks) are shared through a function called *cross-system spool*. Each member of the cluster creates spool files only on spool volumes that it owns, but each member has access to the spool volumes owned by the other members. Users defined as single-configuration virtual machines² have a single logical view of all their spool files. Users logged on to one of the members of

2. Single-configuration virtual machines and multiconfiguration virtual machines are explained in “The user directory” on page 11.

the SSI cluster can access all of their spool files regardless of which member they have logged on to for that session. (For a spool file to be accessible, the member on which it was created must be joined to the cluster.) Users can transfer their spool files to other users or to service virtual machines³ anywhere in the cluster. They can select printers and other spool output devices for processing spool files created on any member of the cluster. Thus, the end-user view of spooling is nearly transparent. Although the CP commands that manipulate spool files operate differently internally for an SSI cluster, additional user options or parameters are not required.

User IDs defined as multiconfiguration virtual machines do not participate in cross-system spool. Each instance of a multiconfiguration virtual machine can view only the spool files owned by the instance on that member. A spool file owned by another instance of the virtual machine (that is, logged on to another member) is not visible. Spool file transfer services, such as RSCS⁴, can be used to move spool files to the member where the intended instance of the multiconfiguration virtual machine resides.

Some important commands that operate on spool files are:

- **SPOOL.** Use the CP SPOOL command to set control options for one or more of your virtual spool devices. A handy way to keep a log of your system activity is to spool your console (SPOOL CONSOLE *, meaning send the console log to yourself), which keeps all your console activity in a spool file. When you close your console (SPOOL CONSOLE STOP CLOSE), your console log is sent to you.
- **QUERY READER ALL.** For class G users, this CP command lets you view information about spool files in your virtual reader. For privileged users, use QUERY READER ALL *userid*, where *userid* is the user ID of a virtual machine. (Without *userid*, a privileged user using this command sees all the reader files on the system.)
- **RDRLIST.** This CMS command displays information about your reader files in a full-screen interactive display.
- **RECEIVE.** This CMS command moves a file from your reader onto a minidisk.
- **PUNCH.** This CMS command punches (copies) a CMS file to your virtual punch.

Related information: For information about managing spool files for the entire z/VM system, start with the summary topics on controlling spool files in *z/VM: System Operation*, SC24-6233:

- Control Spool Files in the Print Queue
- Control Spool Files in the Reader Queue
- Control Spool Files in the Punch Queue

For information about managing spool files for your virtual machine, see “Using Spooled Devices to Print, Punch, and Read Information,” in *z/VM: Virtual Machine Operation*, SC24-6241.

For command help, see *z/VM: CP Commands and Utilities Reference*, SC24-6175, and *z/VM: CMS Commands and Utilities Reference*, SC24-6166.

3. A *service virtual machine* or *service machine* is virtual machine that provides a system service, such as accounting, error recording, or monitoring.

4. IBM Remote Spooling Communications Subsystem (RSCS) Networking, an optional feature of z/VM, is a networking program that enables users on one z/VM system to send messages, files, commands, and jobs to other users within a network.

For online help, type `help` on the CMS command line, then press the Enter key.

The user directory

The z/VM *user directory* (or user registry) describes the configuration and operating characteristics of each virtual machine that can be created by CP. A z/VM user directory exists in two forms: a source form that consists of one or more CMS files, and an object form, created from the source, on a CP-formatted disk.

The source form of the user directory consists of directory statements that define the CP-formatted volume on which the object directory is created and the configuration and operating characteristics of the virtual machines that are known to the z/VM operating system. In an SSI cluster, there are two types of virtual machines:

Single-configuration virtual machine definition

A user ID defined by a single-configuration virtual machine definition (the traditional type of user definition) can be logged on to any member of the SSI cluster, but on only one member at a time. Single-configuration virtual machines are eligible for guest relocation.

Use this type of virtual machine for your Linux guests.

Multiconfiguration virtual machine definition

A user ID defined by a multiconfiguration virtual machine definition can be logged concurrently on multiple members of the SSI cluster. The *instances* of the user ID on the members have common attributes but can also be configured to access different resources.

Multiconfiguration virtual machines are not eligible for guest relocation. Use this type of virtual machine for system support, such as the MAINT user ID, and server virtual machines, such as the TCP/IP server.

The directory statements in the user directory are divided into blocks called *entries*. The entries are:

GLOBALDEFS (global definitions entry)

Which must begin with the GLOBALDEFS directory statement. The entry also includes directory statements that define global settings to be used by all virtual machine definitions.

PROFILE (profile entry)

Each of which begins with a PROFILE directory statement and consolidates other directory statements that are used in common by many virtual machine definitions.

USER (user entry), IDENTITY (identity entry), and SUBCONFIG (subconfiguration entry)

Each of which begins with a USER, IDENTITY, or SUBCONFIG directory statement. USER begins the entry for an single-configuration virtual machine definition. IDENTITY begins the entry for a multiconfiguration virtual machine definition and includes statements common to virtual machine configurations in that definition. SUBCONFIG begins the entry for a set of directory statements in a multiconfiguration virtual machine definition that is specific for a member of an SSI cluster.

The following is a sample single-configuration virtual machine definition. The callouts next to each statement correspond to explanations that follow the sample.

Note: In this document the user directory is modified by using the IBM Directory Maintenance program, DirMaint™, which handles both source and object forms of the user directory. Information about the directory entries is shown for educational purposes only. Unless explicitly instructed to do so, do not attempt to update the user directory source files manually.

```

1 USER LINUXC MYPASS 256M 1G G
2 INCLUDE LINDFLT
3 IPL CMS
4 MACHINE ESA 4
5 CONSOLE 0009 3215
6 NICDEF 4204 TYPE QDIO LAN SYSTEM VSWITCH1
7 SPOOL 000C 3505 A
  SPOOL 000D 3525 A
  SPOOL 000E 1403 A
8 LINK MAINT 0190 0190 RR
  ...
9 MDISK 0191 3390 1595 50 VMLU1A MR
  MDISK 0200 3390 0001 3338 LX1519 MR
  MDISK 0201 3390 0001 3338 LX1559 MR
  MDISK 0202 3390 0001 3338 LX1599 MR

```

1. The USER statement begins the directory entry. The user ID for this virtual machine is LINUXC. “MYPASS” is the user’s logon password. The virtual machine has a default storage of 256 megabytes (“256M”), but you can redefine storage up to a maximum of 1 gigabyte (“1G”). The second “G” means the virtual machine user is a general class user and can control functions for this virtual machine only.
2. The INCLUDE statement specifies the name of a profile entry to be invoked as part of the user entry. In the example, there is a profile entry elsewhere in the user directory that starts with the statement PROFILE LINDFLT.
3. The IPL statement indicates which operating system to load when you log on to the virtual machine. The example shows that CMS will be loaded. Loading CMS is handy because it allows you to make changes to the normal environment as well as run some REXX EXECs (script-like executable files) to set up Linux.
4. The MACHINE statement describes the processor architecture of the virtual machine. The maximum number of virtual CPUs that can be defined for this virtual machine is four. The default is one.
5. The CONSOLE statement defines the operating console (*virtual console*) for the virtual machine. CMS requires console type 3215. If supported by the operating system, you can specify 3270 or issue the CP command `TERMINAL CONSOLE 3270` to change the console from 3215 to 3270.
6. The NICDEF statement defines this virtual machine’s attachment to a z/VM virtual switch.
7. SPOOL statements define the unit record devices. By convention, device number 000C is for the virtual reader (type 3505), device number 000D is for the virtual punch (type 3525), and device number 000E is for the virtual printer (type 1403).
8. LINK statements provide access to another virtual machine’s minidisks.
9. MDISK statements define minidisks owned by the virtual machine. The format of the statement is:

```
MDISK devno type start_cyl extent vol_label access_mode
```

where

devno

Is the virtual device number of the minidisk.

type

Is the disk type of the real disk; typically 3390.

start_cyl

Is the real disk starting location of the first cylinder of the minidisk.

extent

Is the minidisk size in cylinders.

vol_label

Is the volume label of the real disk.

access_mode

Is the access mode. MR means the virtual machine has read/write access.

The following is a sample multiconfiguration virtual machine definition. The callouts next to each statement correspond to explanations that follow the sample.

```
1  IDENTITY OPERATOR pw 32M 32M ABCDEFG
    INCLUDE IBMDFLT
2  BUILD ON MEMBER1 USING SUBCONFIG OPERTR-1
    BUILD ON MEMBER2 USING SUBCONFIG OPERTR-2
    BUILD ON MEMBER3 USING SUBCONFIG OPERTR-3
    BUILD ON MEMBER4 USING SUBCONFIG OPERTR-4
    AUTOLOG AUTOLOG1 OP1 MAINT
    ACCOUNT 2 OPERATOR
    MACHINE ESA
    OPTION MAINTCCW

3  SUBCONFIG OPERTR-1
    LINK OP1 191 192 RR
    MDISK 191 3390 2924 005 FLA293 MR READ WRITE MULTIPLE

    SUBCONFIG OPERTR-2
    MDISK 191 3390 2924 005 FLA2A3 MR READ WRITE MULTIPLE

    SUBCONFIG OPERTR-3
    LINK OP1 191 192 RR
    MDISK 191 3390 2924 005 FLA2B3 MR READ WRITE MULTIPLE

    SUBCONFIG OPERTR-4
    LINK OP1 191 192 RR
    MDISK 191 3390 2924 005 FLA2C3 MR READ WRITE MULTIPLE
```

1. The **IDENTITY** statement begins the directory entry. Like the **USER** statement, the **IDENTITY** statement defines the user ID, password, default storage, maximum storage, and authorization class (or classes) for this virtual machine. You can add statements that are common to all instances of a multiconfiguration virtual machine within the **IDENTITY** entry (**AUTOLOG**, **ACCOUNT**, **MACHINE**, **OPTION**, and so forth).
2. The **BUILD** statement specifies a system name of an SSI cluster member and the **SUBCONFIG** ID of the subconfiguration entry that contains the system-specific directory statements for a multiconfiguration virtual machine instance. In the example, the first **BUILD** statement specifies **MEMBER1** as the cluster member with system-specific directory statements defined in the **OPERTR-1** subconfiguration entry.
If specified, **BUILD** statements must immediately follow an **IDENTITY** statement, an **INCLUDE** statement, or another **BUILD** statement.
3. The **SUBCONFIG** statement starts a subconfiguration entry. This entry contains a set of directory statements in a multiconfiguration virtual machine definition that are specific to one of its virtual machine instances within the SSI cluster. In the example, the **OPERTR-1** subconfiguration entry contains a system-specific **LINK** and **MDISK** statement.

The SUBCONFIG statement must be unique among all users.

Related information: For more information about the user directory, see:

“Creating and Updating a User Directory,” in *z/VM: CP Planning and Administration*, SC24-6178

Overview of the Conversational Monitor System (CMS)

Just as you can interact with Linux or UNIX through a bash or Korn shell, you can interact with z/VM through CMS. Like a shell, you can use CMS to edit files, run EXECs (script-like executable files) or programs, modify the virtual machine environment, or modify z/VM itself. CMS is to z/VM as a shell is to Linux or UNIX.

Minidisks and the CMS access mode

CMS, like other operating systems running in a virtual machine, can access minidisks to store and retrieve files. For CMS, each minidisk has an *access mode* represented by an alphabetic letter that determines how CMS searches for files. In Linux, path variables defining directories determine the search order for files. CMS searches for files among minidisks based on the alphabetical order of the access mode. First, CMS looks on the A minidisk, then the B minidisk, and so forth.

The 191 minidisk holds a special place in CMS. A 191 minidisk to a CMS user is like the home file directory for a Linux user. CMS always tries to access a user's 191 minidisk as access mode A. The CMS 191 minidisk is often called the “A-disk.”

To see your CMS minidisks and their access modes, use the QUERY ACCESSED command. QUERY ACCESSED is similar to the df command in Linux. To access minidisks that are not already in the CMS access order, use the ACCESS command.

Example of viewing and accessing CMS minidisks

1. To view your accessed CMS minidisks, type the QUERY ACCESSED command and press the Enter key:

```
Ready;  
query accessed  
Mode  Stat   Files  Vdev  Label/Directory  
A      R/W     595   191   CHA191  
E      R/O    1776  201   IDT00L  
S      R/O     690  190   CMS21  
Ready;
```

The column under “Mode” shows the access mode for each minidisk. In the example, there are three minidisks accessed as A, E, and S.

Notice that while in CMS all commands end with a “Ready;” prompt, indicating that CMS is ready to do more work.

2. To assign an access mode, use the ACCESS command. For example, to access the minidisk at virtual address 491 as B, type this command and press the Enter key:

```
Ready;  
access 491 b  
DMSACP723I B (491) R/O  
Ready;
```

The response tells you minidisk 491 is accessed read only (R/O) as B.

3. If you assign a mode currently assigned to another minidisk, the new minidisk replaces the current minidisk:

```
Ready;  
access 19d d  
DMSACC724I 19D replaces D (200)  
DMSACP723I D (19D) R/O  
Ready;
```

4. To remove a minidisk from an access mode, use the RELEASE command:

```
Ready;  
release b  
Ready;
```

CMS files

CMS files have a *file name*, *file type*, and *file mode*. File names and file types can be up to 8 characters long. The file mode corresponds to the access mode of the minidisk.

Examples:

```
MYDOC LISTING A1  
DNFPFS LISTPS B1
```

By convention, some file types have special meanings. For example, EXEC is the file type for a file that contains executable statements, LISTING is the file type for text files, and LISTPS is the file type for PostScript files.

To view and manipulate files, use the FILELIST command. FILELIST is similar to the `dir` command in Linux.

Examples of using FILELIST

1. To view all the files on your A-disk, type this command and press the Enter key:

```
filelist
```

Result: You see something like this:

```
CHASTING FILELIST A0 V 169 Trunc=169 Size=253 Line=1 Col=1 Alt=0  
Cmd  Filename Filetype Fm Format Lrecl  Records   Blocks   Date      Time  
CHASTING NETLOG   A0 V      108    2132     53 10/15/03 16:02:30  
KIJLOCMD HGENRPT  A1 V      119     13      1 10/13/03 12:00:40  
KIJLOCMD LOG      A1 V      122    131      2 10/13/03 12:00:37  
KIJLOCMD SCRIPT  A1 V       81    454      4 10/13/03 12:00:37  
REXEC    HELPTCPI A1 V       79    133      2 10/13/03 10:26:11  
NETSTAT  HELPTCPI A1 V       79    749      9 10/13/03 10:25:31
```

2. In the “Cmd” column, you can type commands that are issued against the file on that line.

Example: To edit a file in the filelist, type the XEDIT command in the “Cmd” column:

```

CHASTING FILELIST A0 V 169 Trunc=169 Size=253 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
xedit CHASTING NETLOG  A0 V      108    2132    53 10/15/03 16:02:30
      KIJL0CMD HGENRPT A1 V      119     13      1 10/13/03 12:00:40
      KIJL0CMD LOG     A1 V      122    131      2 10/13/03 12:00:37
      KIJL0CMD SCRIPT  A1 V       81    454      4 10/13/03 12:00:37
      REXEC  HELPTCPI A1 V       79    133      2 10/13/03 10:26:11
      NETSTAT HELPTCPI A1 V       79    749      9 10/13/03 10:25:31

```

3. Use “/” and “=” to avoid extra typing when you enter a command in FILELIST. The “/” means “this file” and “=” can be used to repeat a file name, file type, or file mode.

Example: To copy a file called REXEC HELPTCPI from minidisk A to minidisk D, type this command and press the Enter key (typing over the other columns is OK):

```

CHASTING FILELIST A0 V 169 Trunc=169 Size=253 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
      CHASTING NETLOG  A0 V      108    2132    53 10/15/03 16:02:30
      KIJL0CMD HGENRPT A1 V      119     13      1 10/13/03 12:00:40
      KIJL0CMD LOG     A1 V      122    131      2 10/13/03 12:00:37
      KIJL0CMD SCRIPT  A1 V       81    454      4 10/13/03 12:00:37
copy / = = d HELPTCPI A1 V       79    133      2 10/13/03 10:26:11
      NETSTAT HELPTCPI A1 V       79    749      9 10/13/03 10:25:31

```

4. To see only certain files, use “*” as a wildcard character.

Example: To find any file on any accessed disk with a file type SCRIPT, type this command and press the Enter key:

```
filelist * script *
```

Result: You see something like this:

```

CHASTING FILELIST A0 V 169 Trunc=169 Size=555 Line=32 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
      APLANBD  SCRIPT  A1 V       65      20      1 7/16/02 12:31:01
      APROGBD  SCRIPT  A1 V       80     213      3 7/16/02 12:30:05
      B2HSYS   SCRIPT  Q1 V      113   4910     45 6/17/02 10:42:25
      B2HMSG   SCRIPT  Q1 V       76    670      7 6/17/02 10:42:04
      B2H      SCRIPT  Q1 V       72    107      1 5/20/02 0:47:02
      B2HAPP   SCRIPT  Q1 V       86   3129     25 5/20/02 0:47:02
      B2HEXA   SCRIPT  Q1 V       93   1390     10 5/20/02 0:47:02
      B2HINF   SCRIPT  Q1 V       81   1389     14 5/20/02 0:47:02
      B2HSETUP SCRIPT  Q1 V       70    175      2 5/20/02 0:47:02
      B2HUSE   SCRIPT  Q1 V       89   2622     25 5/20/02 0:47:02
      ACRONYMS SCRIPT  V1 V      962  62886    769 4/05/01 16:27:39
      VMSSERVE SCRIPT  V1 V      103   3180     31 1/24/01 8:48:49

```

The PROFILE EXEC

The PROFILE EXEC is a special executable file analogous to the .profile (or .bash_profile) in Linux and UNIX. Every time a CMS user logs on, CMS runs the PROFILE EXEC residing on the 191 minidisk, file mode A. You can use the PROFILE EXEC to set up your virtual machine environment; for instance, access disks, set up special PF keys, or even load another operating system in your virtual machine.

There can be times when you do not want the PROFILE EXEC to execute when you log on. You can prevent CMS from executing the PROFILE EXEC by issuing **access (noprof**. When you IPL (load) CMS, you see an identifier line displayed and CMS pauses with VM READ in the lower right corner of the display. At that point you can issue **access (noprof**:

```
IPL CMS
z/VM V6.2.0    2004-09-30 16:24
```

```
access (noprof
```

```
VM READ    GDLVME
```

The Help system

z/VM provides online help through the CMS Help system. The HELP command is like the man command in Linux. You can find full descriptions of z/VM commands by using the HELP command. By issuing help, you can access the main help menu for z/VM:

```
HELP TASKS                Task Help Information                line 1 of 40
(c) Copyright IBM Corporation 1990, 2011

z/VM HELP, main panel

The help panels listed below provide information about various
z/VM functions, topics, and tasks.
To view a help panel, move the cursor to any character of the name
or description and press the ENTER key or the PF1 key.

HELP      - Using the z/VM HELP Facility
MENUS     - z/VM help menus
TASKS     - Basic z/VM tasks
AVS       - AVS commands
CMS       - CMS commands
CP        - CP commands
DIRMAINT  - DirMaint commands
DIRMAINT  - DirMaint topics
DUMPS     - Dump commands, subcommands, and utilities
DYNIO     - Dynamic I/O tasks
GLOSSARY  - Definitions of terms
LE        - Language Environment commands
LIBRARY   - z/VM bibliography
MACROS    - CMS assembler macros (menu)
MACROS    - Assembler macros and functions (types)
MESSAGES  - Messages and codes
OPEN      - OpenExtensions services and APIs
PERFKIT   - Performance Toolkit topics
PIPE      - CMS Pipelines stages and subcommands
PIPE      - CMS Pipelines tasks
QUERYSET  - QUERY and SET commands and subcommands
ROUTINES  - CMS callable services (menu)
ROUTINES  - CMS routines (types)
RSCS     - RSCS Networking commands and link parameters
STATEMTS  - REXX, EXEC 2, and EXEC statements
SUBCMDS   - Subcommand groups, such as XEDIT
TCPIP     - TCP/IP commands
PF1= Help   2= Top     3= Quit     4= Return    5= Clocate   6= ?
PF7= Backward 8= Forward 9= PFkeys 10=          11=          12= Cursor

====>

Macro-read 1 File
```

To get quicker access to command information, you can issue the HELP command with one of the keywords you see in the main menu.

Example: For quick access to the information about the CP IPL command, issue:

```
help cp ipl
```

Examples of using the HELP command

1. To get help for all the CP commands, type this command and press the Enter key:

```
help cp menu
```

Result: You see a screen like this:

```
CP MENU                      Menu Help Information                      line 1 of 34
(c) Copyright IBM Corporation 1990, 2011

Help for CP commands

To display a help panel, move the cursor to any character
of the name and press the ENTER key or the PF1 key.
An asterisk (*) preceding the name indicates a menu panel.
A colon (:) preceding the name indicates a task panel.

*CPQUERY  :DUMPS      CPLISTfil DUPLex    LOADBUF   REPeat    SYNCmdrs
*CPSET    :DYNIO      CPRELease Echo     LOADVFCB  REQuest   SYStem
*CPUTIL   :HELP       CPTRAP    ENable    LOCate    RESET     TAg
*DEFINE   #CP         CPTYPE   EXternal LOCATEVM  RESTART   TERMinAl
*DELETE   ACNT        CPU        FLASHBND LOCK      RETAIN    TRace
*DETACH   ACTivate    CPVLoad  FLASHCopy LOGoff    REWInd    TRANSfer
*DISABLE  ASSOCIate    CPXLoad  FLASHEST Logon     SAVESEG   TRSAVE
*DISPLAY  ASTERISK     CPXUnload FLASHRES Message  SAVESYS   TRSource
*DUMP     AT          DEActive FLASHTGT MESSAGEA SCREen    UNCOUPLE
*ENABLE   ATTach      DEDicate FLASHWIT MESSAGEU SEND      UNDEDicate
*INDICATE ATTN        DEFine   Flush     MODify    SET       UNDIAl
*LOCATE   AUTOLog     DEFSEG   FOR       MONitor   SHUTDOWN  UNLOCK
*MODIFY   BACKspace   DEFSYS   FORCE     MSGNOH    SIGNAl    UTILITY
*MONITOR  Begin       DElete   FORWARD  NOTReady  SILENTly  VARY
*PURGE    Change      DESTAGE  FRee     ORDer     SLeeP     VDelete
*RDEVICE  Close       DETach   GIVE      PER       SMsg      VInput
*SPXTAPE  COMmands    Dial     GIVRDEV  POUNDCP   SNAPDUMP  VMDump
*STORE    COMMIT      DISAble  GIVVDEV  PURge     SPACe     VMDUMPTL
*TERMINAL CONCOPI     DISASSOCi HALT     Query     SPool     VMRELocate
*TRACE    COUPLE      DISCARD  HOld     READY     SPXtape   Warning
*TRSOURCE CP          DISConn  HYPerswap RECordin STArt     XAUTOLog
*VMDUMPTL CPACcess    DISPlay  INDicate REDEFine STOP      XLINK
*XLINK    CPCache     DRain    Ip1      REFRESH  STore     XSPool
*XSPool    CPHX       DUmp      LINK

* * * End of File * * *

PF1= Help      2= Top      3= Quit      4= Return      5= Clocate      6= ?
PF7= Backward  8= Forward  9= PFkeys  10=           11=             12= Cursor

====>

Macro-read 1 File
```

2. To get help for a specific command (for example, CP QUERY), type this command and press the Enter key:

```
help cp query
```

Result: You see a screen like this:

```

CP QUERY                      All Help Information                      line 1 of 11
(c) Copyright IBM Corporation 1990, 2011

QUERY

Purpose

You can display various information about your virtual machine by using the
QUERY command operands.

For information on the individual QUERY command operands, press PF11.

* * * End of File * * *

PF1=          2= Top      3= Quit    4= Return    5= Clocate    6= ?
PF7= Backward 8= Forward  9= PFkeys 10=          11= Related   12= Cursor

====>
Macro-read 1 File

```

Related information: For more information about Help, see:

- *z/VM: CMS Primer*, SC24-6172
- For more advanced information, see *z/VM: CMS User's Guide*, SC24-6173.
- For online help, type `help` on the CMS command line, then press the Enter key.

The CMS file editor XEDIT

CMS provides a file editor called XEDIT, which is not only a full-screen editor, but a powerful programming tool. XEDIT has functions similar to `vi` in Linux. This topic introduces you to basic editing functions.

To enter an editing session, use the XEDIT command.

Example: To create a new file called MY FILE A, type this command and press the Enter key:

```
xedit my file a
```

Without any modifications, an editing screen looks like this.

```
MY      FILE      A1 F 80 Trunc=80 Size=0 Line=0 Col=1 Alt=0 1
DMSXIN571I Creating new file: 2

3

4
===== * * * Top of File * * * 5
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7... 6
===== * * * End of File * * *

3

====> 7

8 X E D I T 1 File
```

Numbers in the figure explanations match the reverse type call-outs in the figure:

1. File identification line. The first line displays the file name, file type, file mode and other file characteristics. “F 80” means the length of a line is fixed at 80 characters. “Trunc=80” means any characters beyond the 80-character length are truncated. “Size=0” means there are no lines in this file. “Line=0” means the current line is 0 (more about the current line in point 5). “Col=1” is the position of the column pointer (more about the column pointer in point 6 on page 21). “Alt=0” means the file has had no alterations.
2. Message line. XEDIT communicates with you by displaying messages on the second and third lines.
3. File area. This part of the screen is available to display the file. You can make changes to the file by moving the cursor under any line and typing over the characters, or by using special keys to insert or delete characters. You can make as many changes as you want on the displayed lines before pressing the Enter key. When you press the Enter key, the changes are made to the copy of the file that is kept in virtual storage. The SAVE or FILE subcommand permanently records those changes on the copy of the file that resides on disk. Because a file can be too long to fit on one screen, various subcommands scroll the screen so you can move forward and backward in a file. Scrolling the screen is like turning the pages of a book.
4. Prefix area. The prefix area is the five left-most columns on the screen, and it displays five equal signs (=====). Each line in the file has a prefix area. You can perform various editing tasks such as deleting a line by entering short commands, called prefix subcommands, in the prefix area of a line.
5. Current line. The current line is the file line in the middle of the screen (above the scale). It is highlighted, appearing brighter than the other file lines. The current line is important because most subcommands perform their functions starting with the current line. Naturally, the line that is current changes during an editing session as you scroll the screen, move up and down, and so forth. When the current line changes, the line pointer (not visible on the screen) moves. Many XEDIT subcommands perform their functions starting with the current line and move the line pointer when they are finished.

6. Scale. The scale appears under the current line to help you edit. It is like the margin scale on a typewriter.

The vertical bar (|) in column one on the scale is the *column pointer*. Various subcommands perform their functions within a line starting at the column pointer, which you can move to different positions on the scale by using XEDIT subcommands. The current column is the column under which the column pointer is positioned.

7. Command line. The large arrow (====>) at the bottom of the screen points to the command input area. One way you communicate with the editor is to enter XEDIT subcommands on this line. You can type subcommands in uppercase or lowercase or a combination of both, and many can be abbreviated. For example, BOTTOM, Bottom, and b are all valid ways to type the BOTTOM subcommand (which scrolls the file to the bottom).
8. Status area. The lower right corner displays the current status of your editing session, for example, edit mode or input mode, and the number of files you are editing. The status area in the figure shows you are editing one file.

Tip: If you want to explore XEDIT and its capabilities, type “help” at the XEDIT command line, which opens the XEDIT help menu.

Input mode

By issuing the subcommand INPUT at the command line (you can abbreviate the subcommand as “i”), you enter input mode.

```
MY      FILE      A1  F 80  Trunc=80 Size=9 Line=0 Col=1 Alt=0
DMSXMD573I Input mode:

* * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
- 1

2

====> * * * Input Zone * * * 3                                     Input-mode 1 File
```

XEDIT places the cursor **1** at the beginning of the input zone **2**. The input zone is an area in which you can place data. You can start typing at the cursor and, when you reach the end of a line, press the new line (Enter) key to return the cursor to the beginning of the next line. If you press the new line key on a line without data, XEDIT returns the file to editing mode.

XEDIT blocks the command line **3** because you cannot enter subcommands while in input mode.

Example of using input mode

1. Issue this command:

```
xedit my file a
```

2. On the command line, type **input** and press the Enter key.
3. Type this phrase, then press the Enter key:

```
CP is the z/VM hypervisor
```

4. Type this phrase, then press the Enter key:

```
CMS is the interactive component of z/VM
```

5. Type this phrase, then press the Enter key:

```
XEDIT is the CMS editor
```

6. Press the Enter key.

Result: Your XEDIT screen looks like this:

```
MY      FILE      A1  F 80  Trunc=80 Size=3 Line=3 Col=1 Alt=3
DMSXMD587I XEDIT:

===== * * * Top of File * * *
===== CP IS THE Z/VM HYPERVISOR
===== CMS IS THE INTERACTIVE COMPONENT OF Z/VM
===== XEDIT IS THE CMS EDITOR
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== * * * End of File * * *

=====>

X E D I T  1 File
```

Tip: XEDIT changed all lowercase letters to uppercase. To prevent this, issue the subcommand `set case mixed ignore` before you add text.

Overview of changing files

The simplest way to change a file is to type over text on a line. However, there are times when you want to add or delete data in a file. Special keyboard keys and XEDIT subcommands help you do that:

- **Insert key.** When you press the insert key, the XEDIT cursor changes shape. By placing the cursor on a line, you can insert text between existing letters.
- **Delete key.** By placing the cursor on a line and pressing the delete key, the character to the right of the cursor is deleted and the line closes up.
- **ADD and INPUT prefix commands.** By moving the cursor to the prefix area, typing “a” and pressing the Enter key, you create a new line in the file. If you want to add five lines, type “a5” in the prefix area.
The INPUT prefix command behaves similarly.
- **DELETE prefix command.** By moving the cursor to the prefix area, typing “d” and pressing the Enter key, you delete a line. If you want to delete five lines, type “d5” in the prefix area.
- **LOCATE subcommand.** You can find strings in the file by using the LOCATE subcommand. XEDIT scrolls the file to the line in which the string occurs. The invocation is `1 /string` or simply `/string`.

Example: To find an occurrence of the word “interactive” in a file, issue this command from the XEDIT command line:

```
====> 1 /interactive
```

or simply:

```
====> /interactive
```

Example of changing files

Assume you are still editing the file in “Example of using input mode” on page 22.

1. From the XEDIT command line, type this command and press the Enter key:

```
====> top
```

2. To prevent XEDIT from turning lowercase letters to uppercase, type this command on the command line, then press the Enter key:

```
====> set case mixed ignore
```

3. Move the cursor to the first prefix area, type “a”, then press the Enter key.
4. On the new line, type this phrase:

```
z/VM has many components
```

5. Type over the next line so the letters are in their proper case:

```
CP is the z/VM hypervisor
```

6. Create two blank lines between the first and second lines by typing “i2” in the second prefix area and pressing the Enter key.
7. Delete one of the blank lines by typing “d” in the prefix area, then pressing the Enter key.
8. From the XEDIT command line, locate the word “EDITOR”:

```
====> /editor
```

Result: The line “XEDIT IS THE CMS EDITOR” becomes the current line. You should see a screen like this:

```
MY      FILE      A1  F 80  Trunc=80 Size=5 Line=5 Col=1 Alt=0

===== * * * Top of File * * *
===== z/VM has many components
=====
===== CP is the z/VM hypervisor
===== CMS IS THE INTERACTIVE COMPONENT OF Z/VM
===== XEDIT IS THE CMS EDITOR
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== * * * End of File * * *

=====
```

X E D I T 1 File

SAVE, FILE, QUIT, and QQUIT

SAVE, FILE, QUIT, and QQUIT are XEDIT subcommands:

- Use SAVE when you want to save changes to a file permanently but continue editing the file.

- Use FILE when you want to save changes to a file permanently and quit editing the file.
- Use QUIT to quit editing a file you have not changed. If you have made any changes, XEDIT issues a message:
DMSXSU577E File has been changed; type QQUIT to quit anyway
- Use QQUIT to quit a file and not save any changes you have made since the last save. The subcommand is handy if you decide you do not want any of the changes you have been currently making or if you want to be sure you have not changed a critical file.

Related information: For more information about XEDIT, see:

- *z/VM: CMS Primer*, SC24-6172
- For more advanced information, see *z/VM: XEDIT User's Guide*, SC24-6245.

Summary of Linux and z/VM similarities

Though Linux and z/VM differ in many ways, they have similar functions and commands. Table 1 summarizes the similarities:

Table 1. Linux and z/VM similarities

Linux	z/VM (CP and CMS)
boot	IPL (initial program load)
df command	QUERY ACCESSED command
dir command	FILELIST command
file directory	disk access mode
kernel	Control Program (CP)
man command	HELP command
memory	storage
.profile	PROFILE EXEC
script	EXEC
shell	Conversational Monitor System (CMS)
user registry	user directory
vi	XEDIT

Chapter 2. Planning for Linux virtual servers

This topic covers system requirements and guidelines you need to follow **before** you install z/VM.

The number of Linux virtual servers you need to run depends on many factors. This topic discusses:

- Capacity requirements
- Memory and CPU requirements
- DASD space you need
- Network planning
- User management planning
- z/VM single system image cluster planning
- Obtaining your Linux distribution.

The requirements are first described for a single z/VM system, followed by additional requirements for a z/VM single system image cluster.

Overview of z/VM capacity planning

An important element of z/VM capacity planning is knowing what z/VM is good at: the value of z/VM is its ability to consolidate distributed Linux workloads that under-utilize CPUs or do not require peak processing at the same time. z/VM improves the cost and performance efficiencies because it shares CPU cycles among virtual servers that, if distributed on stand-alone servers, would be idle. There are three key characteristics that you should look for when deciding whether a Linux server could be consolidated on z/VM. Look for Linux workloads that:

- Under-utilize CPUs
- Do not require peak processing at the same time as others
- Have idle times, so that z/VM can share processing cycles with other Linux virtual servers.

Distributed servers running applications being considered for consolidation that run at high utilization throughout the day and peak with other candidate applications are poor candidates for consolidation. In general, the lower the utilization of a candidate application, or the more solitary its peaks are compared to other candidates under consideration, the more likely it can be consolidated.

Likewise, consider a benchmarking strategy that recognizes the real-world characteristics of your Linux workloads. A typical (inappropriate) approach is to conduct atomic measurements that compare throughput of a single instance of an application at a CPU utilization of 100%. This type of benchmarking practice, while simple and easy to conduct, yields inappropriate and misleading expectations of capacity because the practice does not incorporate any of the real-world operational characteristics or highlight any of the elements and advantages of consolidation. While such benchmark comparisons might be appropriate in a distributed paradigm for assessing capacity and performance of stand-alone servers running a single instance of an application, these comparisons are flawed when evaluating z/VM and the mainframe. The flaw is that such comparisons inflate the true operational utilization and throughput of the stand-alone distributed servers and do not account for the ability of z/VM to share idle cycles among virtual servers, which is not possible on under-utilized stand-alone distributed servers. Conducting a benchmark

in such a fashion simply answers the question that, if you had one server running one instance of an application at an assumed utilization of 100%, how much throughput can you expect. In a consolidation case, that is not the question to ask. The question when designing a methodology for assessing capacity for consolidation is how many distributed workloads can you fit on z/VM using the true operational utilization and throughput of those workloads you are considering.

Once you have selected the right set of applications and their servers for consolidation, establish a base set of measurements that capture the real operational throughput of the servers. Figure 5 shows a simplified consolidation example, in which there were many application instances running on separate stand-alone servers. Each of these application servers were 10% busy producing 74 transactions per second.

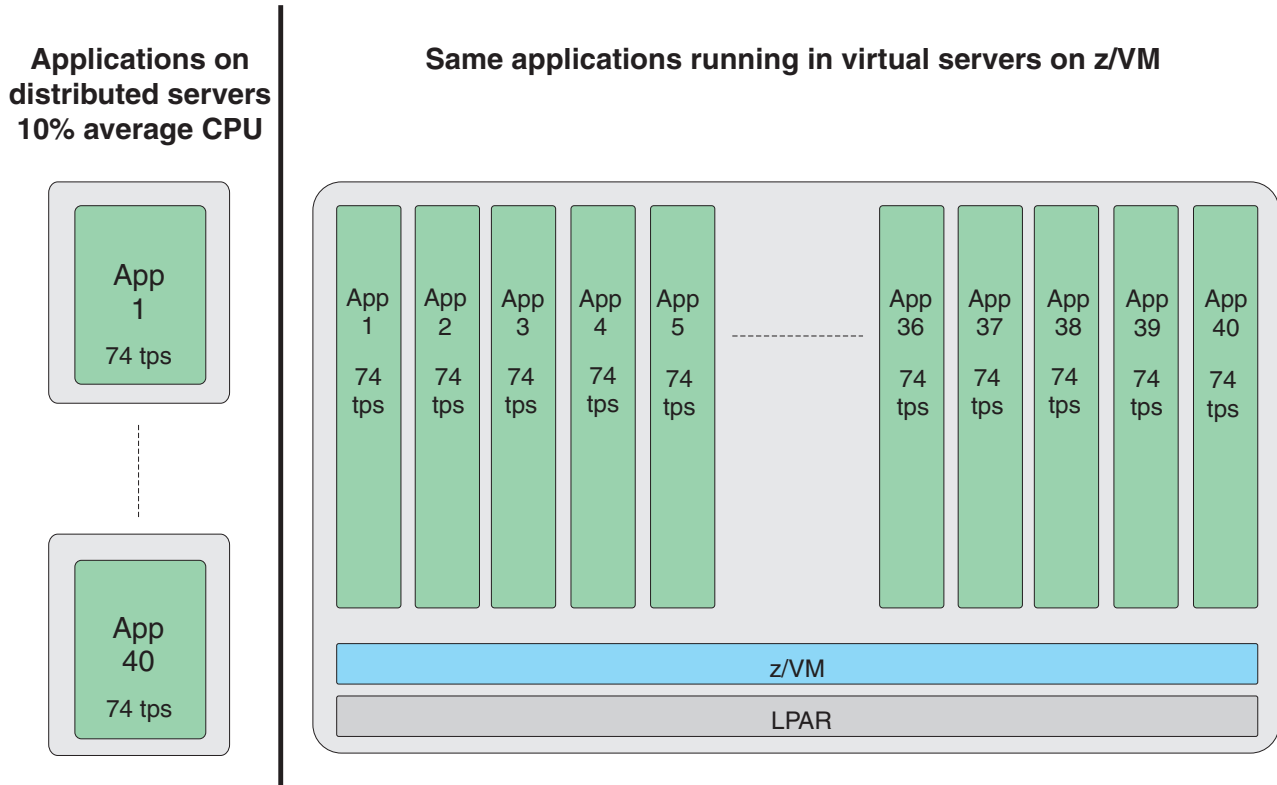


Figure 5. Server consolidation example

When you have established the baseline of 74 transactions per second for the distributed servers, define an equal number of z/VM virtual servers in which to run the applications.

To assess the system capacity required to support the same volume of work, tune the workload driver so that each instance of the application running in a virtual server produces the same transaction rate as its distributed counterpart.

The previous example showed an even distribution of work activity. However, the vast majority of real-world workloads skew the distribution of work. At any given moment, some applications are active while others are less active or idle. Unless your workload is evenly distributed, consider skewing the workload distribution as part of your capacity assessment.

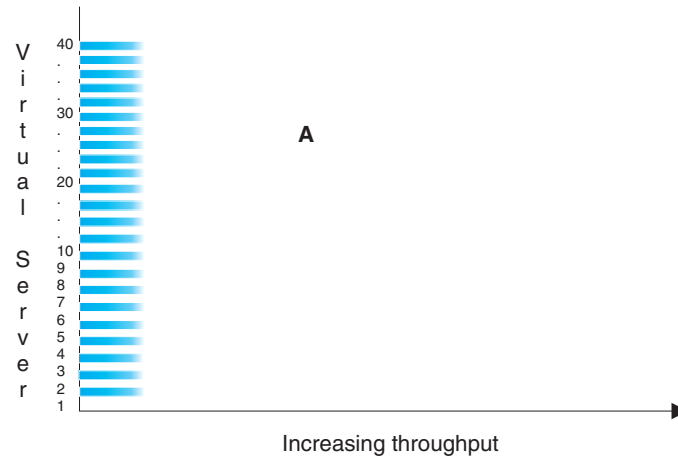


Figure 6. Workload distribution patterns (Part 1 of 3)

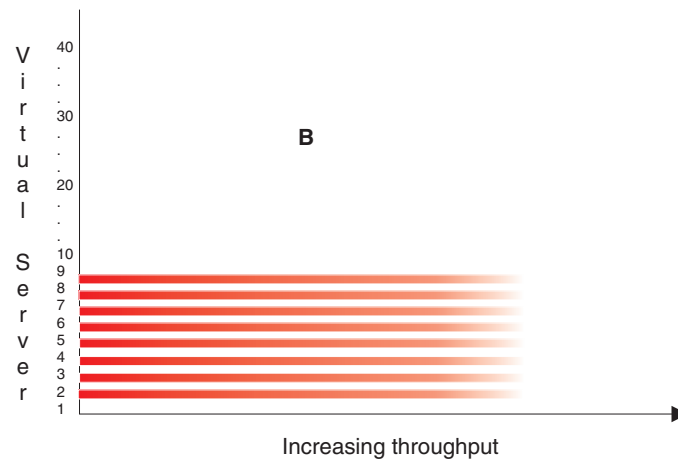


Figure 6. Workload distribution patterns (Part 2 of 3)

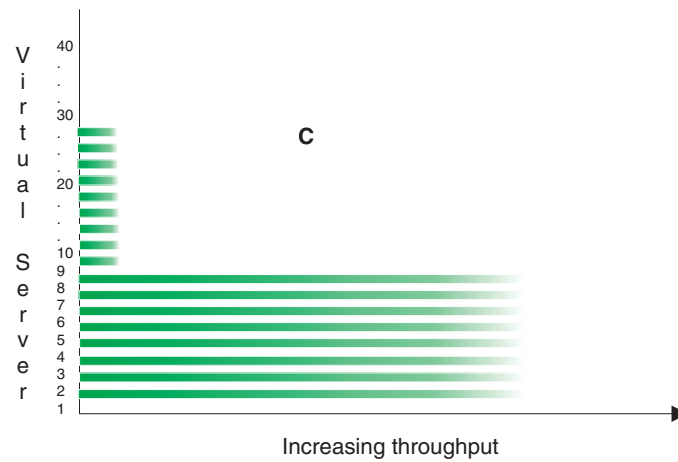


Figure 6. Workload distribution patterns (Part 3 of 3)

Figure 6 shows three workload distribution patterns. Workload distribution pattern A represents the prior example of an even distribution of work activity among the

applications. This pattern shows the worst-case, in which all workloads demand resources at the same time, rather than the characteristics of most production environments. Workload distribution patterns B and C show truer operational characteristics: at any given moment, some applications are busy while others are idle or less busy; and at different times, different applications are busy.

Figure 7 shows the relative throughput capacity of each of patterns A, B, and C.

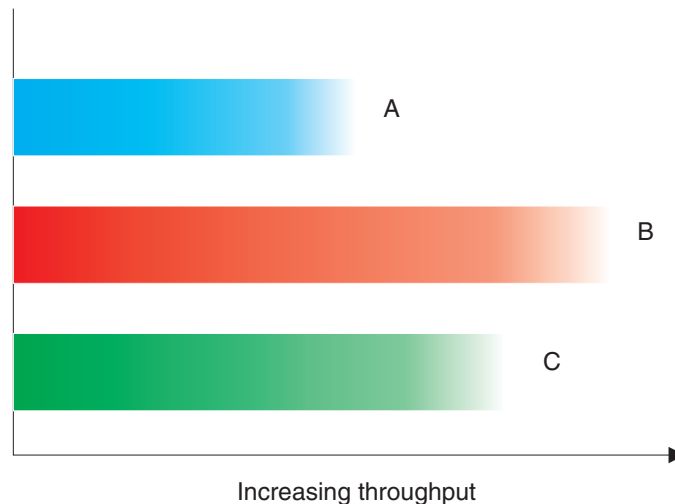


Figure 7. Relative throughput for patterns A, B, and C

Such distributions reflect the real world and place far less stress on the system because they are more cache-friendly and can result in sharply higher capacity results. Likewise, if your workload has characteristics of a skewed distribution, incorporate this aspect into your benchmarking methodology.

Estimating memory and CPU requirements

In most cases, initial system sizings are done with the assistance of IBM, your business partner, or consultant. This section gives you an appreciation for the things considered during an initial sizing and the things you should consider as you add work to your system.

To get you started, this topic gives you some basic knowledge about estimating the memory and CPU requirements for your Linux virtual servers. Such estimating is not an exact science and your experience might vary, but following the guidelines in this topic should help you get started, after which you need to measure the performance and fine tune your system. Topics in Chapter 11, “Monitoring performance and capacity,” on page 147 help you fine-tune your initial configuration.

Overview of estimating memory and CPU requirements

Memory and CPU requirements fall into two categories:

- Real memory and CPU
- Memory and CPU for virtual machines.

Memory for the LPAR

A key factor in determining memory resources is the memory required for your applications. If the applications are new, you must estimate or start at some initial size; you can determine existing application memory requirements if the applications are currently running on other platforms. For example, you might not know how much memory a new WebSphere® application requires, so you can start with 200 MB for the size of the Java Virtual Machine. Additionally, WebSphere Application Server requires 60 MB of memory.⁵ So the total for your new application would be 260 MB. If you have an existing WebSphere application you know requires 250 MB of memory, the total with WebSphere Application Server would be 310 MB.

Include in your estimate the memory requirements for z/VM itself. For information about estimating memory requirements for z/VM itself, see “Host Storage Planning and Administration” in *z/VM: CP Planning and Administration*.

The total memory requirement for your applications, plus memory required for each Linux operating system and z/VM itself, give you an estimate of the memory required for a given LPAR. Do not pad the total memory figure.

Many customers prefer to isolate their applications by running each one in a separate virtual machine. Another strategy is to combine more than one application in a virtual machine, keeping the number of Linux virtual servers to a minimum rather than creating one Linux virtual server for each application. The reason is that each Linux virtual server brings with it some overhead: each Linux operating system itself requires additional memory, and even an idle Linux virtual server uses some CPU resources. Also, applications sometimes can share middleware, which conserves memory. If it conforms to your installation's policy, you might be able to combine more than one application in a virtual machine, thereby saving memory. For example, several WebSphere applications can share the same WebSphere Application Server and the JVM in one virtual machine rather than each application having its own WebSphere Application Server and JVM in its own virtual machine, which would multiply the number of virtual machines and require more total memory.

Of the total memory requirement, try configuring 25% of total storage as expanded storage and the rest to central storage. Many systems do not need more than 2 GB of expanded storage regardless of the total storage available. (Though z/OS no longer supports expanded storage, z/VM and the hardware do.) There are performance advantages to allocating expanded storage. Because z/VM is a 64-bit system, it seems as though allocating all memory to central storage makes sense. However, z/VM manages a paging hierarchy that uses expanded storage first, then slower DASD. Pages move to the slower paging DASD when not referenced within a given time limit. But, if pages are referenced again within the time limit, they can be brought back into central storage rapidly. This paging design gives a more consistent response to users.

During system operations, measure actual memory usage to test the initial memory allocation, which assumes all your virtual servers need the estimated amount of memory all the time. Just as CPU demand has peaks and valleys, so does memory usage.

5. Current requirements as of this writing. For more information about Java and WebSphere memory requirements, consult Java and WebSphere documentation.

Memory for the virtual machines

For the general case of server consolidation, keep the virtual machine size small. How small you can define the virtual machine depends on the applications and workloads running in those virtual machines. Various Linux distributions might have minimum requirements as low as 64 MB. Some applications run fine in those minimum configurations. Other applications and workloads might require larger virtual machines. Avoid defining a virtual machine larger than it needs to be, because Linux uses excess memory for file and buffer caches. On a stand-alone system, these buffers can be very helpful for certain workloads to avoid I/O. However, in a virtual environment, the extra memory consumed adds to overall system memory contention. Such cases could cause a negative impact greater than the positive impact of I/O avoidance, which is especially true in configurations in which data is shared heavily between virtual servers and is mostly read. In those configurations, z/VM minidisk caching can help avoid I/O. As a general guideline, define the virtual machine memory size to keep Linux on the verge of swapping. Lower the memory size until you see Linux begin to swap, then increase the virtual machine memory to the next bigger size.

Another way to reduce the memory requirements is through *discontiguous saved segments* (DCSS). A DCSS is an area of virtual storage outside the address range of a virtual machine. The area can contain read-only data or reentrant code. A DCSS connects discontiguous segments to a virtual machine's address space so programs can be fetched. DCSS sizes can be above 2 GB and combined with Linux to form data areas that are greater than 2 GB in size. DCSSs can be shared by many virtual machines, so total virtual memory required might be reduced. This reduced requirement for virtual memory can:

- Reduce CP paging requirements
- Allow a given z/VM instance to support more Linux virtual machines
- Reduce the amount of Linux disk I/O by having file systems, block devices, and shared objects in DCSSs.

A Linux virtual server machine uses DCSSs through the DCSS block device driver.

Related information: The following are related sources of information.

- This manual shows you how to set up a swap disk on a z/VM minidisk. Other options are available, such as using a virtual disk in storage and DCSS. For more information on swap disk options, see *IBM: Performance Considerations for Linux Guests* (www.vm.ibm.com/perf/tips/linuxper.html).
- For more information about DCSS block device drivers, see *Linux on System z: Device Drivers, Features, and Commands* at:
IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

Real CPU requirements

The real CPU requirement is not simply a function of a single server: rather, the requirement is a function of all your virtual servers combined (see “Overview of z/VM capacity planning” on page 27). You must consider what each of your applications require and then estimate the overall CPU requirement. Consider the processor share each virtual server receives for satisfactory performance (see “Steps for using CP commands to improve performance” on page 165). Your IBM representative, business partner, or consultant offer services to help you perform this task.

When defining z/VM LPARs, typically assign a minimum of two logical processors (many middleware products advise that you use two logical processors). You have the option of dedicating the processors to the LPAR or sharing them with other LPARs. It is also possible to set different processing weights to LPARs, which gives more processor resources to one over another. For instance, you might entitle your production LPAR to 60% and your test LPAR to 40% of the processor resources. For details, consult the *Processor Resource/Systems Manager™ Planning Guide* for your System z server.

Virtual CPU requirements

In general, follow this guideline: define as many virtual CPUs as needed (maximum CPU resources required), but do not exceed the number of real processors assigned to this LPAR. Extra virtual CPUs just add to the overhead and potentially increase the software multiprocessing factors.

If the Linux program `cpuplugd` is available, define the maximum number of virtual CPUs that a Linux server could use. `cpuplugd` can tailor the virtual CPUs actually used by the Linux server, and thus cause no additional CP overhead. If `cpuplugd` is not available, do not define more virtual CPUs than the Linux server is likely to use.

Guidelines for estimating the amount of DASD you need

Here are some guidelines on estimating the amount of DASD you need.

Estimating DASD for z/VM paging

- Provide sufficient paging DASD for the paging subsystem. As shipped, z/VM has one full paging volume, which might be sufficient for your system. A rule of thumb is to provide twice as much DASD space as your total virtual storage requirement. This value can be decreased by a fraction of the real memory available. See “Paging Space” in *z/VM: CP Planning and Administration*.
- Add paging space on a volume basis: do not use the paging volume for other purposes.
“Steps for adding a paging, spooling, temporary disk, or user volume” on page 49 tells you how to add a paging volume.

Related information: For information about how z/VM uses DASD space and DASD space calculations, see “Direct Access Storage Requirements” in *z/VM: CP Planning and Administration*, SC24-6178.

Estimating DASD for the Linux file system

- For disks for each Linux virtual server, the simplest guideline is to provide one 3390-3 at a minimum. A one-volume configuration supports many of the default packages installed by each Linux distribution. The need for additional DASD space depends on which additional products you install, the levels of those products, the size of user applications associated with those products, and end-user data.

Linux views disk space in bytes while z/VM views disk space according to the device geometry (for example, number of cylinders). Regardless of the 3390 model, each cylinder holds approximately 670 KB when formatted. A 3390-3 has 3338 usable cylinders, which means it can hold about 2.2 GB when formatted. The difference between 3390 models is the number of cylinders the model has. A 3390-9 has three times the capacity of a 3390-3, so it holds approximately 6.6 GB.

Consult product documentation for additional DASD space requirements. Your Linux distributor will tell you how much disk space to use for minimum and full installations.

- Eliminating certain Linux packages can help reduce your DASD requirements. Consider a minimal Linux installation to save disk space.
- Another strategy to save DASD is to divide up the Linux file system into read-only and read/write minidisks, then share the read-only minidisks among the virtual machines running Linux.
- Do not dedicate DASD to Linux (that is, do not allocate an entire DASD to Linux for its exclusive use). Instead, use minidisks for Linux volumes, especially those whose I/O operations are primarily read operations, to take advantage of minidisk caching. The z/VM Control Program by default maintains a minidisk cache for better performance. However, if there are many write operations to the minidisk, the extra Control Program overhead used to maintain the cache outweighs the benefits of caching. If the I/O operations include many write operations, turn off minidisk caching for the minidisk by using the CP command SET MDCACHE MDISK or the MINIOPT statement in the user directory.
- The Parallel Access Volume facility allows a controller to offer multiple device numbers that resolve to the same DASD, which allows I/O to the same DASD to happen concurrently. If you do not have the Parallel Access Volume facility, each DASD can do only one I/O at a time. To have your Linux file system perform well without the Parallel Access Volume facility, you can spread the file system across separate volumes and use the Linux logical volume manager, which maximizes the opportunity for the I/Os to the volumes to happen concurrently. For example, you could set up two stripes so Linux can do two I/Os at a time. Note that the volumes should be on separate controllers to avoid contention.

A feature of IBM DASD subsystems, IBM HyperParallel Access Volume (HyperPAV) allows multiple I/O operations to a DASD through a pool of base and alias subchannels that are shared within a logical subsystem (LSS). With HyperPAV, z/VM eliminates the need for users to map volumes to aliases and takes care of the aliases and I/Os automatically.

Planning your network

To use Linux, you need to connect to your TCP/IP network.

You need to determine:

- The device addresses of your network interface.
- The host name and domain name of your Linux virtual servers
- The IP address and the subnet mask
- Depending on the connectivity type, the broadcast name server and network addresses.
- Whether you plan to have your Linux virtual servers use the cryptographic facility for SSL acceleration.

TCP/IP networking options for Linux

z/VM provides two broad categories for TCP/IP connectivity for Linux virtual servers:

- Real network interfaces with connections to the LAN. A real network connection might be through any device supported by Linux, including IBM Open Systems Adapters and channel-attached devices. The real device (as defined in the Input/Output Configuration Data Set) must be dedicated to the virtual machine running Linux. You can do this by providing DEDICATE entries in the CP directory entry for the virtual machine or by using the CP ATTACH command.

- Virtual network interfaces, which allow the real connections to be shared, maximizing throughput. Virtual network connections include:
 - Guest LAN. Through a guest LAN, z/VM simulates OSA-Express or HiperSockets™ microcode to allow you to connect virtual server systems to communication adapters. Such connections enable virtual servers to communicate through a LAN rather than through point-to-point connections. If the virtual servers require external connectivity, that connection requires a virtual machine acting as a router between the guest LAN and the external connection.
 - Virtual switch. A virtual switch is a special kind of guest LAN. In addition to providing a network of virtual adapters, the switch can be connected directly to an OSA-Express QDIO adapter. This capability allows you to gain connectivity to external LAN segments without requiring a router, reducing CPU utilization and latency associated with providing external connectivity through a router.

The virtual switch is the preferred way to connect your Linux machines to the network, but there are other legacy connection types. For more information, see “Networking Options in z/VM” in *z/VM: Connectivity*, SC24-6174.

Figure 8 on page 36 is a diagram of a virtual switch called VSWITCH1. Coupled to the virtual switch through NICDEF directory statements are Linux virtual servers. The DTCVSW1 service virtual machine, running a subset of the TCP/IP stack, is the virtual switch controller. The full TCP/IP stack runs in the TCPIP service virtual machine, the z/VM production TCP/IP. The two service virtual machines are kept separate so you can operate them independently.

Due to the advantages of virtual switches, this document shows you how to set up a virtual switch configuration only. IBM created the TCP/IP and user directory changes for the default virtual switch controllers DTCVSW1 and DTCVSW2⁶, but there are other tasks you must do. These tasks are intermingled with other configuration tasks in Chapter 3, “Changing the system configuration,” Chapter 5, “Configuring TCP/IP,” and Chapter 7, “Creating your first Linux virtual machine and installing Linux.” To place the tasks in context, Table 2 summarizes how to configure TCP/IP and a virtual switch.

Table 2. Task roadmap for configuring TCP/IP and a virtual switch

Subtask	Associated instructions (see . . .)
Define a virtual switch for z/VM	“Steps for defining a virtual switch and defining equivalency IDs for OSA devices” on page 63
Configuring the production TCP/IP	“Setting up the production TCP/IP” on page 83
Configuring TCP/IP to be logged on automatically	“Steps for automatically starting TCP/IP” on page 83
Connecting your virtual machine to the virtual switch	“Overview of defining virtual machines for Linux” on page 93
Configuring Linux to use the virtual switch	Your Linux installation documentation. For Linux, the connection is defined as an OSA-Express device.

6. A second virtual switch controller is available for failover: CP automatically chooses another virtual switch controller should the first one fail.

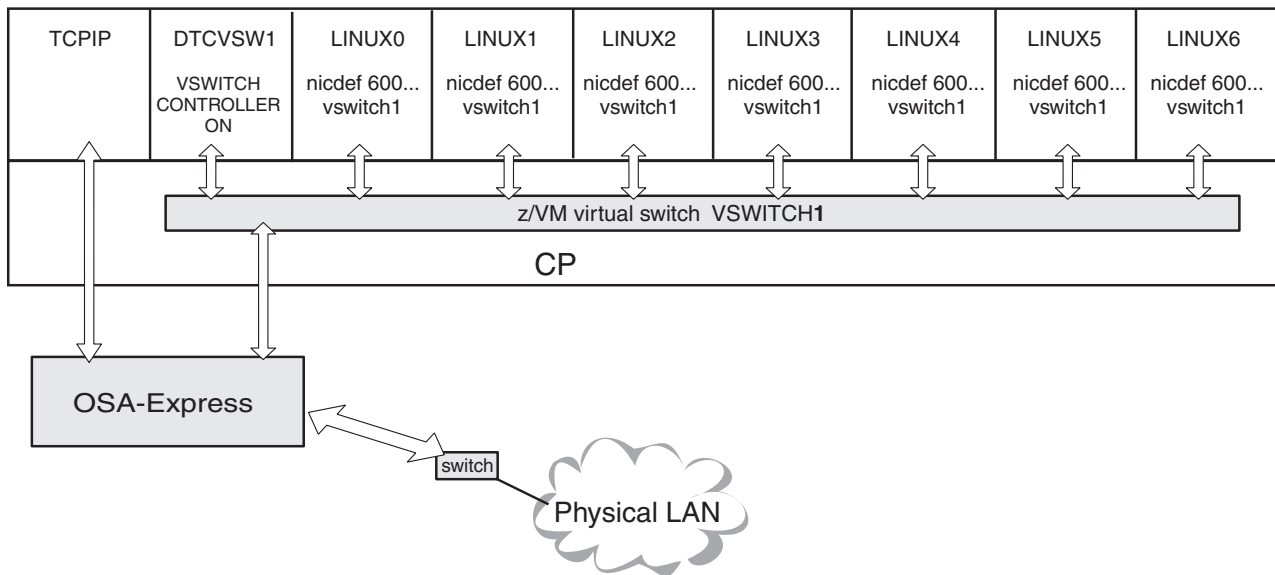


Figure 8. Example of a virtual switch

Related information: For more information, see “Networking Options in z/VM” in *z/VM: Connectivity*, SC24-6174

Giving Linux virtual servers access to cryptographic hardware for SSL acceleration

To run Linux as a virtual server under z/VM with access to cryptographic hardware for SSL acceleration, you must:

1. Define the cryptographic facility for the LPAR in which z/VM runs through the Hardware Configuration Definition.
2. Define the cryptographic capability for each Linux virtual machine in the user directory.
3. Have the z90crypt device driver integrated into the Linux operating system. Some distributions have the device driver integrated, while other distributions require you to install it.

The user directory statement CRYPTO APVIRT provides access to the cryptographic hardware and allows the z90crypt device driver to use cryptographic instructions. z/VM manages a pool of hardware cryptographic queues that are shared among all the virtual servers using the cryptographic facility. You can create more virtual servers that share the cryptographic facility than the actual number of hardware queues available. Even though the hardware queues are shared, the data remains isolated and is not vulnerable or exposed to other Linux images.

“Steps for defining a master virtual machine for Linux” on page 93 shows you how to add the CRYPTO APVIRT user directory statement to the master Linux virtual machine, which means all replicas of this master have access to the cryptographic facility. If you prefer, you can leave this statement out of the master Linux virtual machine and add the user directory statement to individual Linux virtual machines only.

z/VM provides CP commands to manage the cryptographic facility. See “Step for managing real devices” on page 122, and “Virtual machine operation tasks” on page 137.

Related information: The following are additional sources of information:

- For more information about defining the cryptographic facility for the LPAR in which z/VM runs, consult your hardware and Hardware Configuration Definition documentation.
- For more information about z/VM's support for the cryptographic facility, see "Cryptographic Acceleration" in *z/VM: CP Planning and Administration*, SC24-6178.
- For information about setting up secure SSL communications, see "Configuring the SSL Server" in *z/VM: TCP/IP Planning and Customization*, SC24-6238.
- For information about the z90crypt device driver, see *Linux on System z: Device Drivers, Features, and Commands* at:

IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

Planning for user management

To add a new user to z/VM, you must create a directory entry for a new virtual machine. Through native facilities, you can update a file called USER DIRECT, then run the DIRECTXA utility to compile the source file and place the new user directory online. The USER DIRECT file is simply a CMS file containing various directory statements. A virtual machine definition is a grouping of directory statements beginning with a USER or IDENTITY statement and ending with either the next USER or IDENTITY statement or the end of the file. (For a description of virtual machine types and the USER and IDENTITY statements, see "The user directory" on page 11.)

You can administer the user directory by editing the USER DIRECT file, then placing the user directory online through the DIRECTXA command. However, such a method of user management is cumbersome and error prone. Because the user directory is so important for z/VM, a corrupt or invalid online user directory can be disastrous. For example, if you inadvertently overlap minidisk definitions, it could cause serious data loss. If your z/VM system has more than a handful of virtual machines, it makes little sense to manage users manually. You need an automated facility.

The Directory Maintenance Facility for z/VM (often called DirMaint) is just such an automated facility. DirMaint is a CMS application that helps you manage your z/VM user directory through a simplified command interface and automated facilities. DirMaint commands, which are like their corresponding directory statements, initiate user directory transactions. DirMaint error checking ensures that only valid changes are made to the user directory, and that only authorized personnel are able to make the requested changes. Any transaction requiring the allocation or deallocation of minidisk extents can be handled automatically. You can control all user-initiated transactions through passwords and record transactions for auditing purposes.

When you activate DirMaint, you give control over the user directory to the DIRMAINT service virtual machine. The source USER DIRECT file on the PMAINT virtual machine's 2CC disk is no longer valid and you must not use the DIRECTXA command. DirMaint maintains and updates the online user directory. You interact with the DIRMAINT service machine through commands to make changes to the user directory.

A handy DirMaint function for virtual machines is the capability to define template (or prototype) directory entries. With this function, you can clone a virtual machine with a few commands by instructing DirMaint to create a new virtual machine like

the prototype entry and to replicate the prototype's disks. For more information about using templates for cloning virtual machines, see Chapter 8, “Cloning Linux virtual servers,” on page 105.

DirMaint also has a service virtual machine called DATAMOVE. The service machine moves the contents of CMS-formatted minidisks from one disk to another, then erases the contents of the CMS minidisks being deleted. Because these functions are time consuming, the functions are offloaded from the DIRMAINT service machine to the DATAMOVE service machine, making DIRMAINT available to process commands.

DirMaint is preinstalled on your system in a disabled state. To use DirMaint, you must first pay a license fee, then enable the feature and configure it. Table 3 gives you an overview of the tasks involved for enabling and configuring DirMaint.

Table 3. Task roadmap for enabling and configuring DirMaint

Subtask	Associated instructions (see . . .)
Enabling the Directory Maintenance Facility	“Steps for enabling DirMaint” on page 71
Configuring the Directory Maintenance Facility	<ul style="list-style-type: none"> • “Steps for adding DirMaint maintenance and changing the passwords for the DirMaint service machines” on page 72 • “Steps for configuring DirMaint” on page 73 • “Steps for authorizing users to perform DirMaint tasks” on page 75 • “Steps for controlling where DirMaint creates minidisks” on page 75 • “Steps for putting the configuration into production and starting DirMaint” on page 79
Testing and using the Directory Maintenance Facility	<ul style="list-style-type: none"> • “Steps for testing DirMaint” on page 80 • “Step for modifying the OPERATOR's directory entry” on page 81

Related information: For more information, see:

- *Program Directory for IBM z/VM Directory Maintenance Facility Feature* at: IBM: z/VM Internet Library (www.ibm.com/vm/library/)
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190

Planning for single system image clusters and live guest relocation

Previous planning topics considered z/VM as a single system. This topic describes additional planning guidelines for a single system image (SSI) cluster and live guest relocation.

General guideline: As a preferred practice, define the same real device numbers to reference the same devices on all members of the SSI cluster. You need to discuss this practice with your hardware administrator to ensure the IOCDs reflects this.

Guidelines for LPAR, memory, and processors in an SSI cluster

By following these requirements and guidelines before you set up your SSI cluster, you avoid disruptions due to the need to add memory or processing power.

- For ease of installing hardware and software service, it is best to install no more than two members of an SSI cluster on the same mainframe.
- Additional processor and memory requirements in an SSI cluster (compared to a single system) might be necessary to relocate virtual servers. You might need “whitespace” (additional processor and memory capacity) for relocating virtual servers.

DASD guidelines and setup in an SSI cluster

SSI clusters allow DASD to be either shared or nonshared (member specific). CP recognizes volume ownership, which must be set for each CP-owned volume. As depicted in Figure 9 on page 40, DASD volumes in an SSI cluster comprise:

- One set of cluster-wide volumes (shared). The common volume (default label VMCOM1) contains shared data files for the SSI cluster, such as the SSI *persistent data record (PDR)*, which holds information about the state of each member of an SSI cluster. This volume also contains the cluster-wide minidisks.
- One set of release volumes for each z/VM release in the cluster (shared). Service for release *vrn* (where *vrn* is a 3-digit string that identifies the z/VM version, release, and modification level) is loaded to minidisks on one or two release-specific volumes (default labels *vrnRL1* and *vrnRL2*).
- One set of system volumes for each member (nonshared and shared). The system residence (IPL) volume is member-specific and therefore nonshared. The default volume label is *M0mRES*, where *m* is the number of the member in the member list on the SSI configuration statement (for example, *M01RES* for member 1). This volume contains member-specific data such as the warm start and checkpoint areas, the object directory, and the standard system minidisks.

The CP-owned volumes for paging and temporary disks are also member-specific and nonshared. The default volume labels are *M0mPnn* and *M0mTnn*, respectively (for example, *M01P01* and *M01T01* for member 1).

Paging storage should be twice the size (or more) than the total virtual storage of all virtual servers. Remember to include any virtual servers that might relocate to a system when adding up the total virtual storage for that system.

- The spool volumes owned by each member are shared with the other members. Each member creates spool files only on its own spool volumes, but each member can manipulate spool files on other member's spool volumes. Single configuration virtual machines must be able to manipulate their spool files no matter which member they are logged on to; however, multiconfiguration virtual machines can manipulate only the spool files created on the member they are currently logged on to. Thus, spool volumes are owned by a particular system, but all systems in the SSI cluster must see them and treat them as spool volumes.

Figure 9 on page 40 shows the spool volumes as owned by each member, but shared. The default spool volume labels are *M0mSnn* (for example, *M01S01* for member 1 and *M02S01* for member 2).

- User (PERM) volumes can be shared or nonshared. Figure 9 on page 40 shows three shared user volumes: *LINUSR1*, *LINUSR2*, and *LINUSR3*. The figure also shows each member with one nonshared user volume (for example, Member 1 has *M01U01*).

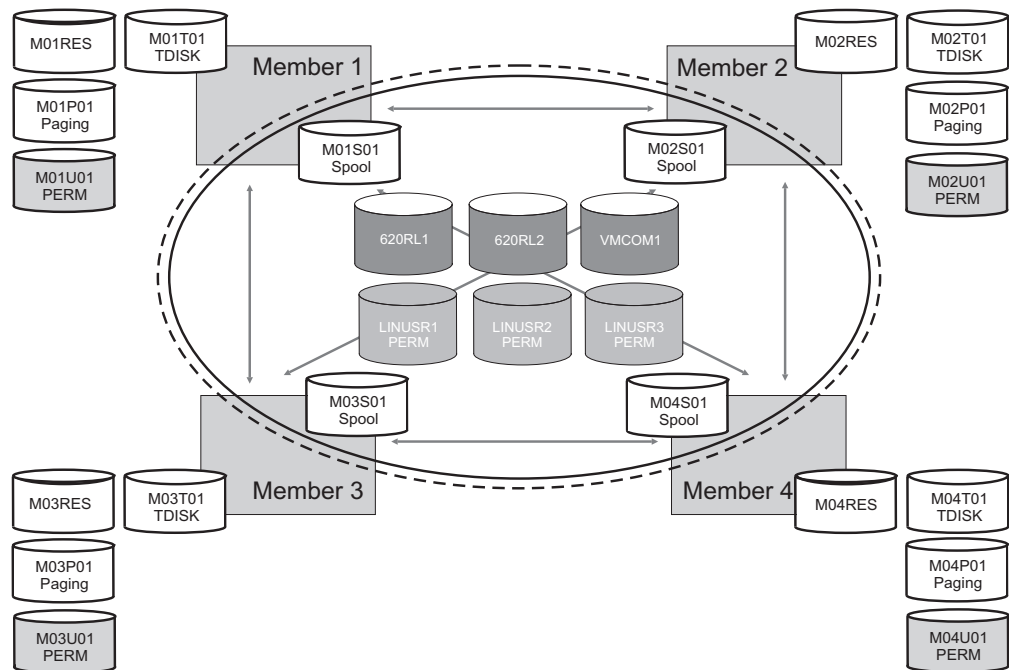


Figure 9. DASD volumes in an SSI cluster

The following are requirements for relocating virtual servers using minidisks or dedicated DASD volumes:

- DASD must be shared between the source and destination systems.
- For devices that identify themselves uniquely, such as FICON-attached DASD, CP assigns an equivalency ID number (EQID) to the DASD at IPL time. This EQID is compared when relocations are attempted.
- For SCSI or FCP-attached DASD, you must assign an EQID for other DASD on each system. You can assign an EQID dynamically through the SET command, or statically in the SYSTEM CONFIG file (see Chapter 3, “Changing the system configuration,” on page 49).

Related information: For more information about the equivalency ID, see descriptions of the EDEVICE and RDEVICE statements in “The System Configuration File” in *z/VM: CP Planning and Administration*, SC24-6178.

Guideline for saved segments in an SSI cluster

To relocate successfully virtual servers that use a saved segment (named saved system or discontinuous saved segment), the saved segments on all systems in the SSI cluster must match exactly. Build your saved segments from disks shared throughout the cluster, not disks private to one member. Build the saved segments on one system and use SPXTAPE to copy the saved segments to the other systems in your cluster. For information about SPXTAPE, see *z/VM: CP Commands and Utilities Reference*.

Guidelines for planning your network in an SSI cluster

Each member in the SSI cluster must have a direct ISFC connection to every other member in the SSI cluster. In other words, SSI traffic from one member to another never flows through an intermediate member. Figure 10 on page 41 illustrates this topology with no intermediaries, sometimes called “fully connected.”

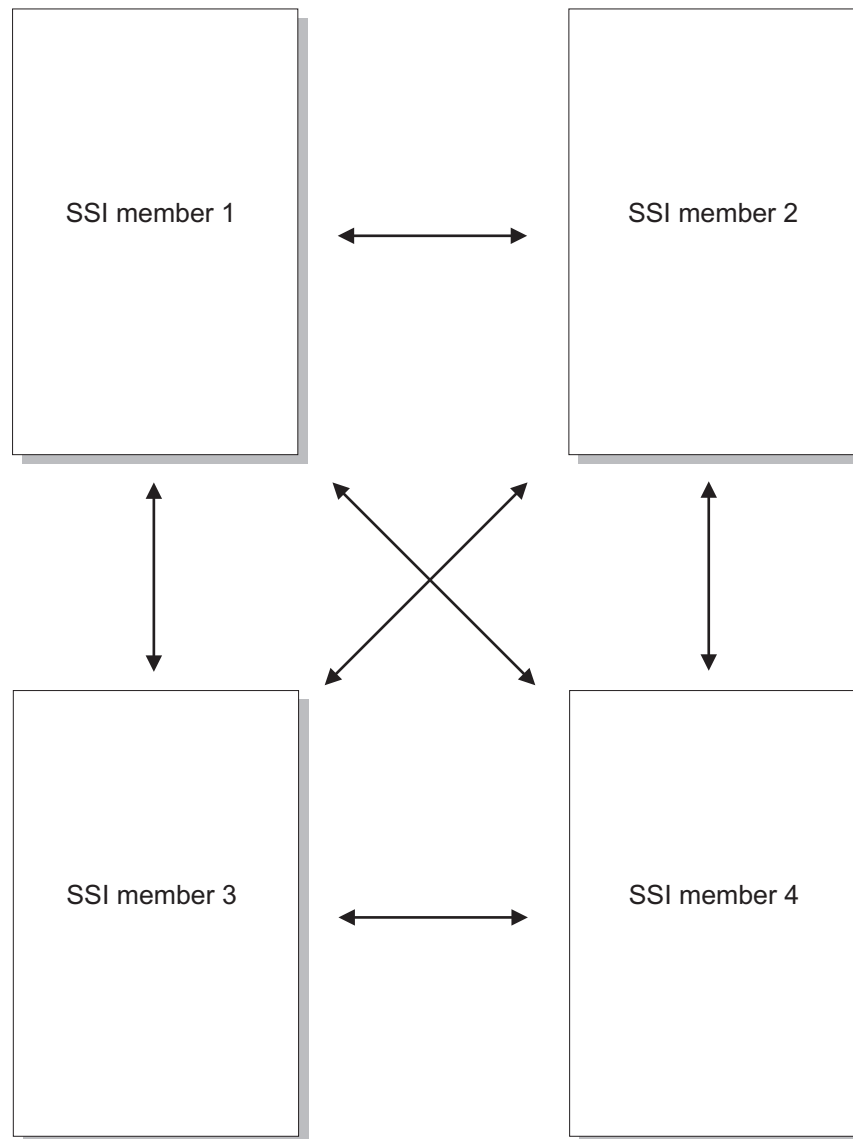


Figure 10. Fully connected ISFC topology

The ISFC connection from one member to another is called an *ISFC logical link*, or simply a *logical link*. There is always exactly one ISFC logical link between two members. Figure 11 on page 42 illustrates that there is exactly one ISFC logical link connecting member 1 to member 2.

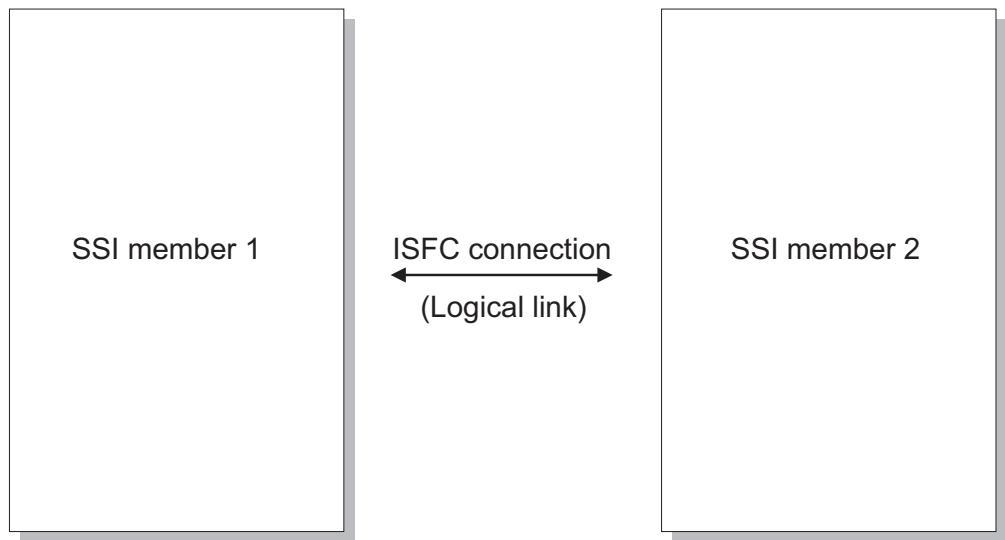


Figure 11. Logical link

A logical link is composed of 1 - 16 channel-to-channel (CTC) devices. The channel paths on which these CTCs reside can be either unswitched channel paths or switched channel paths. For example, the logical link might be composed of eight CTCs spread among two unswitched FICON channels. Figure 12 illustrates the example, depicting four CTCs on one FICON channel and four more CTCs on another FICON channel. Together these eight CTCs compose the logical link.

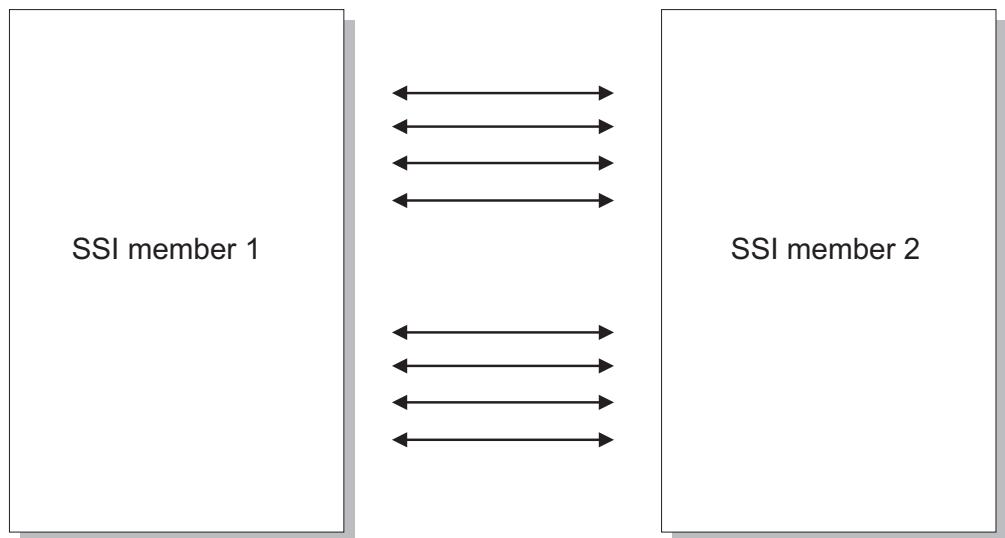


Figure 12. One logical link composed of multiple CTC devices

Follow these guidelines for your initial SSI configuration:

- Use FICON channels of the same speed; for example, all FICON channels at either 2 Gb/sec, 4 Gb/sec, or 8 Gb/sec. If you need to create a logical link with FICON channels of differing speeds, see “Planning the ISFC Network in an SSI Cluster” in *z/VM: CP Planning and Administration*.
- For each logical link, use four FICON channels. If four channels are too many for your environment, you can use as few as one. However, be aware that the fewer

channels you use, the lower the capacity of the link. Also, be aware that if you use only one channel, you have no redundancy in case of a hardware failure.

- Configure four real CTC devices on each FICON channel, using the same real device numbers on both ends of the link. For example, device 6000 on member 1 should connect to device 6000 on member 2.

Planning your virtual server networks

The following are guidelines for LAN and SAN connections:

- For shared LAN connections, follow these guidelines:
 - In order to relocate virtual servers, the source and destination systems must be connected to the same physical LAN segment.
 - OSA, HiperSockets, and LCS devices must be assigned an EQID on the source and destination systems. You can do this dynamically through the SET command or statically in the SYSTEM CONFIG file. The devices assigned the same EQID must be equivalent devices.
 - OSA or HiperSockets devices dedicated to virtual servers must be assigned an EQID on the source and target member.
- For shared Storage Area Network (SAN) connections, follow these guidelines:
 - In order to relocate virtual servers, the source and destination members must be connected to the same SAN fabric.
 - FCP devices must be assigned an EQID on the source and destination members. You can do this dynamically through a SET command or statically in the SYSTEM CONFIG file. The devices assigned the same EQID must be equivalent devices.

Cryptographic requirements in an SSI cluster

The following are requirements for relocating virtual servers using the cryptographic feature:

- Relocation is supported only for virtual servers using APVIRT cryptography.
- The same feature types must be available on the destination as on the source member. For example, a virtual server that uses CEX2A features on the source cannot relocate to a member that has only CEX3A features.

Tape requirements in an SSI cluster

The following are requirements for relocating virtual servers that use tapes:

- In order to relocate virtual servers, the source and destination members must be connected to the same physical tape drives.
- For FICON attached tapes, CP assigns an EQID to the tape at IPL time. This EQID is compared when relocations are attempted. For other tapes, an EQID must be assigned on the source and destination members. You can do this dynamically through the SET command or statically in the SYSTEM CONFIG file. The devices assigned the same EQID must be equivalent devices.

Guideline for user management in an SSI cluster

For the SSI to function properly and relocation to proceed properly, the user directories on all members of an SSI cluster must remain synchronized. An SSI cluster should be set up with a directory manager; for example, the Directory Maintenance Facility for z/VM (DirMaint).

DirMaint uses several server virtual machines to ensure that the user directory is current on all members in the cluster. The DIRMAINT and DATAMOVE machines

run on one member, while DIRMSAT n and DATAMOV n server machines run on the other members, one set per member. For instance, MEMBER2 has DIRMSAT2 and DATAMOV2, MEMBER3 has DIRMSAT3 and DATAMOV3, and so forth. Updates to one member are propagated to all members through communication among the server virtual machines.

Steps for planning your virtual servers

The following task is a simplified approach for planning the resources for a z/VM system. The approach presents the concepts of planning, but is not a replacement for full capacity planning practices.

Before you begin: Read the previous guidelines for a single z/VM system and an SSI cluster.

Determine names and device requirements for all the Linux virtual servers in your cluster.

Perform these steps to plan for your SSI cluster:

1. Enter the name of your Linux virtual server in the left column of Table 4 on page 46.
2. Determine a set of applications that will run in that Linux virtual server. The set might include one or more applications that process a given workload. For example, if you use a WebSphere application, you need to include the WebSphere application server and the JVM.
3. Determine the total memory requirement for the application set.
4. Add the memory requirement for the Linux operating system to the application set memory requirement. Enter the result in the “Memory” column in Table 4 on page 46.
5. Determine the virtual server requirements for processors and devices. Fill in the corresponding columns in Table 4 on page 46.
6. In the “Member” columns, mark where the virtual server will normally reside and where the virtual server can relocate.
7. Repeat steps 1 to 6 for all your virtual servers.

You are done planning your virtual servers. Continue with “Steps for planning your SSI cluster.”

Steps for planning your SSI cluster

Before you begin: Complete “Steps for planning your virtual servers.”

Perform the following steps to plan your SSI cluster:

1. Determine the maximum number of virtual servers you might need to run on each member. Fill in the “Maximum number of resident and relocated virtual servers” row in Table 4 on page 46.

The table helps you plan where virtual servers need to relocate if any one system is down for maintenance or if an entire System z mainframe is down for maintenance. Keep in mind that you might want a virtual server to be able to run on several different systems so you can apply hardware or software service without disruption.

2. Determine the maximum amount of memory needed (normal resident virtual servers, plus all virtual servers that might be relocated) on each system. Fill in the “Maximum memory for normally resident and relocated virtual servers” row in Table 4 on page 46.
3. In Table 4 on page 46, fill in the “Memory for z/VM”. Include the size of virtual disks in storage (VDISKS) in that memory requirement.
4. Calculate the “Total virtual memory” requirement.
5. If the maximum memory is greater than the capacity of a system, prioritize your virtual servers. For instance, virtual servers used for test or non-critical functions might be shut down during service outages rather than relocated. Subtract the memory requirement for any virtual servers that could be shut down during maintenance windows.
6. Determine the real memory requirement after considering overcommitment and fill in the row “Total real memory requirement (after considering overcommitment).” Typically, the total virtual memory should not be more than three times the real memory requirement.

It is common to overcommit memory in a z/VM environment. The question is how much you can overcommit without system performance becoming unacceptable. How much you can overcommit depends on many factors, such as workload, tolerance of service level agreements, and paging configuration. Determining an acceptable level of overcommitment is beyond the scope of this document. If you do choose to overcommit, adjust your calculations in this step based on that level of overcommitment.
7. Determine the amount of central and expanded storage needed according to the formulas in Table 4 on page 46.
8. Determine the number of real CPUs you require and record the number in Table 4 on page 46.
9. Calculate the DASD paging space and enter it in the “DASD paging space” row in Table 4 on page 46. The calculation should be two (or more) times the total virtual memory requirement. This calculation helps you determine the DASD volumes needed for step 10.
10. Plan your system names, DASD addresses, and CTC devices for your real systems. See “Complete the tape installation worksheets” in *z/VM: Installation Guide*.

You are done planning your SSI cluster. Save Table 4 on page 46 to help you with your virtual server relocation plan in Chapter 13, “Preparing for live guest relocation,” on page 169.

Next, follow the installation instructions in *z/VM: Installation Guide*.

Related information: Additional guidance is in “Creating a z/VM SSI Cluster” in *z/VM: CP Planning and Administration*, SC24-6178.

46

z/VM: Getting Started with Linux on System z

Steps for obtaining documentation and media

You must obtain your Linux distribution documentation and installation media.

Before you begin: You need to decide which Linux distribution you will use.

Perform these steps to obtain documentation and media:

1. Obtain Linux documentation according to your distribution:
 - SuSE SLESLinux documentation at:
SUSE Linux Enterprise Server Trial (www.novell.com/products/server/eval.html)
 - Red Hat, Inc., documentation at:
Red Hat Enterprise Linux (www.redhat.com/rhel/server/mainframe/)
2. Access the installation media provided with your distribution. You need access to:
 - The Linux system files (kernel, parm file, and initial RAM file)
 - A UNIX-based FTP or NFS server attached to your network.

You are finished when you have obtained the documentation and access to installation media.

Chapter 3. Changing the system configuration

This topic tells you how to change the z/VM systems (members) in your SSI cluster by updating the SYSTEM CONFIG file. As a system programmer, you should become familiar with the SYSTEM CONFIG file.

The *SYSTEM CONFIG* file contains the primary system definitions used when CP is booted (IPLed). All of the information needed to configure CP statically comes from this file. In an SSI cluster, all members use the same SYSTEM CONFIG file; however, you can specify that certain configuration statements apply only to specific members by qualifying the statements with a system identifier. This topic has examples of this specifying method.

The SYSTEM CONFIG file resides on a special CMS-formatted minidisk (CF0) belonging to the PMAINT user ID. Minidisks containing such objects are called parm disks because when allocated those disks are given a special record category type called "PARM". There can be more than one parm disk allocated in a z/VM system for backup and recovery.

Related information:

"Using Configuration Files" in *z/VM: CP Planning and Administration*, SC24-6178.

Steps for adding a paging, spooling, temporary disk, or user volume

z/VM allocates DASD volumes according to their use. In this topic, you format and allocate paging, spooling, temporary disk, and user volumes. A *paging volume* (PAGE allocation) is a volume owned by CP that is used by the paging subsystem; a *spooling volume* (SPOL allocation) is a volume owned by CP that is used by the spooling subsystem for spool files; a *temporary disk volume* (TDSK allocation) is a volume on which disk space is allocated to the user only when the user is logged on; a *user volume* (PERM allocation) is a DASD volume that CP uses for permanent minidisks.

In an SSI cluster, paging, temporary disk, and some user volumes are unique to each system (nonshared), while spooling and other user volumes are shared throughout the cluster. You should assign an owning system to paging, temporary disk, and nonshared user volumes to avoid data integrity issues. CP ensures that only the owning system writes to these volumes. Spooling volumes need a particular system to be the primary owner, even though they are shared in the SSI cluster. Shared user volumes are assigned to the SSI cluster.

Before you can attach a volume, you must use the CPFMTXA utility program to format, label, allocate, and set ownership for the volume. Later, after you do these tasks, you will add the volumes to the appropriate places in SYSTEM CONFIG.

Before you begin: You need to know how many paging, spooling, temporary disk, and user volumes you need. See "Guidelines for estimating the amount of DASD you need" on page 33 and "DASD guidelines and setup in an SSI cluster" on page 39.

You need to log on as MAINT. You need to have one or more real DASD volumes connected to your mainframe.

Perform these steps to prepare a paging, spooling, temporary disk, or user volume:

1. Attach the volume to MAINT.

Example: If the volume is at real address 202, type this command and press the Enter key:

```
attach 202 *  
DASD 0202 ATTACHED TO MAINT 0202 WITH DEVCTL  
Ready;
```

2. To gain access to the CPFMTXA utility, access the 551 minidisk as F. From the command line, type the following command and press the Enter key:

```
access 551 f  
Ready;
```

3. Start CPFMTXA:

```
cpfmtxa 202 label
```

where *label* is a volume label. For instance, you might have the following labels for volumes connect to SSI member MEMBER01:

- M01P01 for a paging volume
- M01S01 for a spooling volume
- M01T01 for a temporary disk
- M01U01 for a user volume.

4. Type in these responses and press the Enter key in response to each prompt:

```
CPFMTXA:  
FORMAT WILL ERASE CYLINDERS 000000000-000003338 ON DISK 0202  
DO YOU WANT TO CONTINUE? (YES | NO)  
yes  
:  
ENTER INPUT COMMAND:  
  
END  
  
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0  
ENTER ALLOCATION DATA  
:  
:
```

5. In response to the “ENTER ALLOCATION DATA” prompt, type one of these responses and press the Enter key:

- If you are creating a paging volume:

```
page 0 end  
end
```

- If you are creating a spooling volume:

```
spo1 0 end  
end
```

- If you are creating a temporary disk volume:

```
tdsk 0 end  
end
```

- If you are creating a user volume:

```
perm 0 end  
end
```

Result: You see more messages from the program. Finally, you see this:

```
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

6. Set the ownership for the volume:

```
cpfmtxa 202 label owner ssiname.sysname
```

where:

label

is a volume label. For instance, on SSI member MEMBER01, you might have the following labels:

- M01P01 for a paging volume
- M01S01 for a spooling volume
- M01T01 for a temporary disk
- M01U01 for a user volume.

ssiname

is the SSI cluster name.

sysname

is the SSI member name that owns the device.

Note: For paging, temporary disk, and spooling volumes, specify *ssiname.sysname*. For user volumes that are shared throughout the SSI cluster, specify NOSYS for the *sysname* (OWNER *ssiname.NOSYS*). For user volumes that are not shared throughout the SSI cluster, specify the *sysname* (OWNER *ssiname.sysname*).

7. Detach the volume.

```
detach 202  
DASD 0202 DETACHED  
Ready;
```

8. Repeat these steps for other paging, spooling, temporary disk, and user volumes you need.
9. Record the volume label and allocation type for each volume you have defined. You will use this information later when updating the CP-owned volume list and user volume list.

You are done for now. Later, you will make these volumes available to CP for its use.

Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG

Before you begin updating the SYSTEM CONFIG file, you must access the parm disk.

Before you begin: You need to log on as MAINT.

Perform these steps to access the parm disk:

1. Access the parm disk (PMAINT's CF0). Type these commands and press the Enter key after each command:

```
link pmain cf0 cf0 mr
Ready;
access cf0 z
Ready;
```

2. As a preferred practice, create a backup of SYSTEM CONFIG:

```
copy system config z system configbk z
```

You know you are done when you have access to the CF0 disk and have created a backup of SYSTEM CONFIG on the parm disk.

Steps for updating the CP-owned volume list

The *CP-owned volume list* is the place where you specify the labels of paging, spooling, and temporary disk volumes that CP should automatically attach to the system during IPL. These volumes contain the system data for your z/VM system. All other volumes on the system are considered user volumes.

Before you begin: You need to format and allocate your paging, spooling, and temporary disk volumes. See “Steps for adding a paging, spooling, temporary disk, or user volume” on page 49.

You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG.”

Perform these steps to update the CP-owned volume list:

1. Edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Result: You see a file like the following:


```

/*****
/*      SYSTEM CONFIG FILE                               */
/*****
/*      */
/* Refer to CP Planning and Administration for SYSTEM CONFIG rules */
/*      */
/* Warning - Always run CPSYNTAX after updating the SYSTEM CONFIG */
/*      */
/*****

/*****
/*      System_Identifier Information                       */
/*****

      System_Identifier LPAR LP01 MEMBER1
      System_Identifier LPAR LP02 MEMBER2
      System_Identifier LPAR LP03 MEMBER3
      System_Identifier LPAR LP04 MEMBER4

/*****
/*      SSI Statement required for VMSSI feature          */
/*****

      SSI MYSSI   PDR_VOLUME VMCOM1,
                SLOT 1 MEMBER1,
                SLOT 2 MEMBER2,
                SLOT 3 MEMBER3,
                SLOT 4 MEMBER4

```

2. Find the string “CP_Owned Slot 13” in the “DUMP & SPOOL VOLUMES” section. At the XEDIT command line, type this command and press the Enter key:

```
====> /cp_owned/ & /slot/ & /13/
```

3. Add additional spooling volumes for the other SSI members to slots in ascending order (slots 11, 12, 13, and so forth).
 - a. On the XEDIT command line, type **input** and press the Enter key.
 - b. Add additional CP_OWNED statements for the spooling volumes. Press the Enter key after adding each statement.
 - c. When finished, press the Enter key once more.

Example: If you are adding a spooling volume to each member in a four-member SSI cluster, you might see the following:

```

CP_Owned  Slot 10  M01S01
CP_Owned  Slot 11  M02S01
CP_Owned  Slot 12  M03S01
CP_Owned  Slot 13  M04S01
CP_Owned  Slot 14  M01S02
CP_Owned  Slot 15  M02S02
CP_Owned  Slot 16  M03S02
CP_Owned  Slot 17  M04S02

```

4. Find the string “CP_Owned Slot 255” in the “PAGE & TDISK VOLUMES” section. At the XEDIT command line, type this command and press the Enter key:

```
====> /cp_owned/ & /slot/ & /255/
```

- a. In the prefix area, type **i** on the line with the BEGIN statement.

Example:

```
00123 /*****  
00124 /* Page and Tdisk volumes for Member 1 */  
00125 /*****  
00126  
00127 MEMBER1: BEGIN  
00128 CP_Owned Slot 255 M01P01  
00129 MEMBER1: END
```

- b. Type in a new CP_OWNED statement using the volume label of a paging volume for the first member that you allocated in “Steps for adding a paging, spooling, temporary disk, or user volume” on page 49. Add additional paging volumes to slots in descending order (slots 254, 253, 252, and so forth) for the first SSI member.

Example:

```
00123 /*****  
00124 /* Page and Tdisk volumes for Member 1 */  
00125 /*****  
00126  
00127 MEMBER1: BEGIN  
00128 CP_Owned Slot 254 M01P02  
00129 CP_Owned Slot 255 M01P01  
00130 MEMBER1: END
```

5. Repeat step 4 on page 53 for the other SSI members.

Note: For nonshared volumes such as temporary disk, paging, and system residence volumes, you can use the same slot numbers for each SSI member as long as you qualify the CP_OWNED statements for each member. Shared volumes, such as spooling volumes, must all use the same slot on each member.

6. Follow steps 4 on page 53 through 5 to add temporary disk volumes to each SSI member. Add the temporary disk volumes to slots in descending order from the next available slot that was not used previously.
7. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```

/*****
/*                                DUMP & SPOOL VOLUMES      */
/* Dump and spool volumes begin with slot 10 and are assigned in */
/* ascending order, without regard to the system that owns them. */
/* All member systems MUST have access to ALL spool volumes, and they */
/* MUST be assigned the same slot number on each system. Volume */
/* ownership, which determines which member owns and can allocate */
/* the spool space, must have been previously established by CPFMTXA. */
*****/

CP_Owned  Slot  10  M01S01
CP_Owned  Slot  11  M02S01
CP_Owned  Slot  12  M03S01
CP_Owned  Slot  13  M04S01
CP_Owned  Slot  14  M01S02
CP_Owned  Slot  15  M02S02
CP_Owned  Slot  16  M03S02
CP_Owned  Slot  17  M04S02

/*****
/*                                PAGE & TDISK VOLUMES      */
/* Paging and T-disk volumes are not shared. To avoid interference */
/* with spool volumes and to automatically have all unused slots */
/* defined as "Reserved", begin with slot 255 and assign them in */
/* descending order. It is NOT required that each system use the */
/* same slot numbers for paging and T-disk volumes. Volume ownership */
/* must have been previously established by CPFMTXA OWNER. */
*****/

```

```

/*****
/* Page and Tdisk volumes for Member 1 */
*****/

MEMBER1: BEGIN
    CP_Owned Slot 252 M01T02
    CP_Owned Slot 253 M01T01
    CP_Owned Slot 254 M01P02
    CP_Owned Slot 255 M01P01
MEMBER1: END

/*****
/* Page and Tdisk volumes for Member 2 */
*****/

MEMBER2: BEGIN
    CP_Owned Slot 252 M02T02
    CP_Owned Slot 253 M02T01
    CP_Owned Slot 254 M02P02
    CP_Owned Slot 255 M02P01
MEMBER2: END

/*****
/* Page and Tdisk volumes for Member 3 */
*****/

MEMBER3: BEGIN
    CP_Owned Slot 252 M03T02
    CP_Owned Slot 253 M03T01
    CP_Owned Slot 254 M03P02
    CP_Owned Slot 255 M03P01
MEMBER3: END

/*****
/* Page and Tdisk volumes for Member 4 */
*****/

MEMBER4: BEGIN
    CP_Owned Slot 252 M04T02
    CP_Owned Slot 253 M04T01
    CP_Owned Slot 254 M04P02
    CP_Owned Slot 255 M04P01
MEMBER4: END

```

Steps for activating ISFC links between SSI members

ACTIVATE ISLINK statements identify channel-to-channel adapters (CTCAs) for ISFC links between the members of the cluster. The installation procedure defines two links between each member, but you might want additional links for better throughput and faster guest relocations. See “Guidelines for planning your network in an SSI cluster” on page 40.

Before you begin: You need access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform the following steps to activate ISFC links:

1. Find the string “ACTIVATE ISLINK” in the section titled “Activate ISLINK statements.” At the XEDIT command line, type this command and press the Enter key:

```
====> /activate islink
```

2. At the XEDIT command line, type this command and press the Enter key:

```
====> input
```

3. Add additional ACTIVATE ISLINK statement pairs (one for each end of the link).

Example:

```
MEMBER1:  ACTIVATE ISLINK 14A0 14A1  NODE MEMBER2
MEMBER2:  ACTIVATE ISLINK 14A0 14A1  NODE MEMBER1

MEMBER1:  ACTIVATE ISLINK 14B0 14B1  NODE MEMBER2
MEMBER2:  ACTIVATE ISLINK 14B0 14B1  NODE MEMBER1
```

4. Press the Enter key and add up to, but no more than, four ACTIVATE ISLINK statements for each SSI member.
5. When you are done, press the Enter key once more.
6. Save the file.

You should now see something like the following:

```
MEMBER1:  ACTIVATE ISLINK 14A0 14A1  NODE MEMBER2
MEMBER2:  ACTIVATE ISLINK 14A0 14A1  NODE MEMBER1

MEMBER1:  ACTIVATE ISLINK 14B0 14B1  NODE MEMBER2
MEMBER2:  ACTIVATE ISLINK 14B0 14B1  NODE MEMBER2
```

Steps for updating the user volume list

Just as there is a list of DASD volumes that CP should automatically attach to the system during IPL for access to CP system areas, there is a list of DASD volumes that CP should automatically attach to the system for user minidisk definitions. Because all minidisks are managed by CP, all volumes that house minidisks must be attached to the z/VM system. CP must control the volumes so it can reorient channel programs initiated by a guest operating system. The guest perceives its disks as starting at cylinder 0, but the true location of the guest's minidisk starts at an offset of real cylinder 0.

Unless you qualify user volume statements with a system name, the user volumes are attached to all members in the SSI cluster.

If no user volumes are attached to the system at IPL time, the real devices housing minidisks need to be attached manually (see “Step for managing real devices” on page 122). Otherwise, virtual machines will have no disks. To avoid manual attachment, you can tell CP to look for DASD volume labels and attach those devices at IPL time.

The `USER_VOLUME_LIST` statement directs CP to attach specific user DASD volumes at z/VM load (IPL) time. The `USER_VOLUME_INCLUDE` statement allows you to create a general volume identifier and to include all volumes that match the general identifier.

Example: If all your Linux user volumes have a volume identifier starting with `V2LX`, you can add this statement:

```
User_Volume_Include V2LX*
```

Tip: If a volume is normally attached to the system using a `USER_VOLUME_INCLUDE` statement, CP does not notify the operator if the volume is not mounted. If a user volume is necessary for normal system operation, specify it with a `USER_VOLUME_LIST` statement so that the operator is notified during system initialization if the volume is not mounted.

Before you begin: You need to format and allocate the user volumes you need. See “Steps for adding a paging, spooling, temporary disk, or user volume” on page 49.

You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for `SYSTEM CONFIG`” on page 52.

If not already doing so, you must edit the `SYSTEM CONFIG` file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to update the user volume list:

1. Find the string “`User_Volume_List`” in the section titled “User volumes for local minidisks.” At the XEDIT command line, type this command and press the Enter key:

```
====> /user_volume_list
```

2. For shared user volumes, update the `User_Volume_List` statements in the “Shared User Volumes” section.

Example:

```
00204 /*****  
00205 /* Shared User Volumes */  
00206 /*****  
00207         User_Volume_List LINUSR1  
00208         User_Volume_List LINUSR2  
00209         User_Volume_List LINUSR3  
00210         User_Volume_List LINUSR4
```

3. In the prefix area for the last `USER_VOLUME_LIST` statement, type “`i`” and press the Enter key. For instance:

```
00204 /*****  
00205 /* Shared User Volumes */  
00206 /*****  
00207         User_Volume_List LINUSR1  
00208         User_Volume_List LINUSR2  
00209         User_Volume_List LINUSR3  
i0210         User_Volume_List LINUSR4
```

4. Add other User_Volume_List statements as needed, one per line.

Example:

```
00204 /*****  
00205 /* Shared User Volumes */  
00206 /*****  
00207         User_Volume_List LINUSR1  
00208         User_Volume_List LINUSR2  
00209         User_Volume_List LINUSR3  
00210         User_Volume_List LINUSR4  
00211         User_Volume_List LINUSR5
```

5. Move the cursor after the User_Volume_List statements. In the prefix area, type “i” and press the Enter key.
6. Add User_Volume_Include statements.

Example:

```
User_Volume_Include V2LX*
```

7. For unshared user volumes, update each member section with user volumes for that member. For instance:

```
/*  
/* User volumes for Member 1 */  
/*  
MEMBER1: BEGIN  
         User_Volume_List M01U01  
MEMBER1: END  
/*  
/* User volumes for Member 2 */  
/*  
MEMBER2: BEGIN  
         User_Volume_List M02U01  
MEMBER2: END
```

8. When you finish the User_Volume_Include statements, press the Enter key.
9. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```

/*****
/* Shared User Volumes
*****/
User_Volume_List LINUSR1
User_Volume_List LINUSR2
User_Volume_List LINUSR3
User_Volume_List LINUSR4
User_Volume_List LINUSR5
User_Volume_Include V2LX*
/*****
/* User volumes for Member 1
*****/
MEMBER1: BEGIN
User_Volume_List M01U01
MEMBER2: END
/*****
/* User volumes for Member 2
*****/
MEMBER2: BEGIN
User_Volume_List M02U01
MEMBER2: END

```

Steps for updating the FEATURES statement

The FEATURES statement in SYSTEM CONFIG allows you to modify attributes associated with the running system at IPL time. In this procedure, you will modify some of the features:

- The Auto_Warm_IPL feature causes CP to bypass prompting for start options, provided the previous system shutdown was successful. The feature allows for a fully automated startup of z/VM.
- The Clear_TDisk feature causes CP to erase temporary disks fully (that is, overwrite the entire temporary disk with zeros) after those disks are detached. The feature prevents another user who might define an identically sized temporary disk from accessing data written by the previous user.
- The Retrieve defines the default and maximum number of retrieve buffers allowed per user on your system. Retrieve buffers create a command history, from which users can retrieve commands previously issued. Command retrieval is usually assigned to a program function key such as PF12 (F12). The assignment is through the CP SET command, SET PF12 RETRIEVE. By pressing PF12, a command is retrieved and written back into the command area on the terminal screen. You probably do not need to change these settings.
- The Passwords_on_Cmds feature tells CP whether to prompt users for passwords when using the CP AUTOLOG, LINK, or LOGON commands.
- The Disconnect_timeout feature controls whether and when a virtual machine is logged off after it has been forced to disconnect. You will turn this feature off, so that any virtual machine that has been forced to disconnect will not be logged off.
- The ShutdownTime and Signal ShutdownTime features enable a virtual machine to register with CP to receive a shutdown signal when z/VM is shutting down (see “Steps for enabling Linux virtual servers to shut down automatically” on page 109). CP waits to shut itself down until the time interval (in seconds) is exceeded, or all of the virtual machines enabled for the signal shutdown have reported a successful shutdown. Some Linux distributions support this function, which allows Linux to shut down cleanly before z/VM shuts down.

Note: Unless you qualify the features with a system name, these features will be the same for all members in your SSI cluster.

Before you begin: You need access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to update the FEATURES statement:

1. Find the line containing the text “Set_Privclass” in the section titled “Features Statement.” At the XEDIT command line, type this command and press the Enter key:

```
====> /set_privclass
```

2. At the command line, type this command and press the Enter key:

```
====> input
```

3. Type this line (indent three spaces), then press the Enter key twice:

```
    Enable ,
```

You might also change the comments for “Auto_Warm_IPL” and “Clear_TDisk.”

4. Under “Passwords_on_Cmds ,”:
 - a. Change the three instances of “yes” to “no.” You might also want to change the comments for these lines to reflect your change.
 - b. Move the cursor to the line containing the text “Logon no.” After the word “no” place a space and a comma.
5. Add a comma at the end of “Vdisk Userlim 144000 blocks.”
6. Move the cursor to the prefix area on the line “Vdisk Userlim 144000 blocks ,”, type “i4” followed by a space, then press the Enter key.
7. Type these features (indent as shown) on the new lines:

```
    Disconnect_timeout off
    Set ,
      ShutdownTime 30 ,
      Signal ShutdownTime 500
```

8. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```

SYSTEM  CONFIG  Z1  F 80  Trunc=80 Size=258 Line=148 Col=1 Alt=4

Features ,
  Disable ,                      /* Disable the following features */
    Set_Privclass ,              /* Disallow SET PRIVCLASS command */
  Enable ,
    Auto_Warm_IPL ,              /* Bypass prompting at IPL          */
    Clear_TDisk ,                /* Clear TDisks when detached       */
  Retrieve ,                      /* Retrieve options                  */
    Default 20 ,                 /* Default.... default is 20        */
    Maximum 255 ,                /* Maximum.... default is 255      */
  MaxUsers noLimit ,              /* No limit on number of users      */
  Passwords_on_Cmnds ,            /* What commands allow passwords? */
    Autolog no ,                 /* ... AUTOLOG does not            */
    Link no ,                    /* ... LINK does not               */
    Logon no ,                   /* ... and LOGON does not, too     */
  Vdisk Userlim 144000 blocks ,   /* Maximum vdisk allowed per user */
  Disconnect_timeout off
Set,
  ShutdownTime 30 ,
  Signal ShutdownTime 500

```

Steps for controlling access to devices at startup

Sometimes your z/VM system might have access to devices that you do not want to be varied online during IPL. For instance, the devices might duplicate labels of devices used by your production system, or might be in use by other LPARs or systems. You can specify ranges of devices that z/VM should not vary online during IPL.

Before you begin: You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to control access to devices at startup:

1. Find the line containing the text “Status of Devices.” At the XEDIT command line, type this command and press the Enter key:

```
====> /status of devices
```

2. On the line with the string “Devices,” type **i** in the prefix area and press the Enter key:

```

00304 /*****
00305 /*                               Status of Devices                               */
00306 /*****
00307
i00308 Devices ,
00309   Online_at_IPL  0000-FFFF,
00310   Sensed        0000-FFFF

```

3. Type the following line, then press the Enter key:

```
Offline_at_IPL 0000-FFFF,
```

4. Type over the device range on the ONLINE_AT_IPL statement with a device range specific for your system.

Example:

```
Online_at_IPL 1000-1fff,
```

5. Add a new ONLINE_AT_IPL statement for each device range. Use the **i** prefix command to add lines.

Example:

```
00304 /*****  
00305 /*                               Status of Devices                               */  
00306 /*****  
00307  
00308 Devices ,  
00309     Offline_at_IPL 0000-FFFF,  
00310     Online_at_IPL  1000-16ff,  
00311     Online_at_IPL  1800-18ff,  
00312     Online_at_IPL  2000-20ff,  
00313     Sensed       0000-FFFF
```

6. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```
SYSTEM  CONFIG  Z1  F 80  Trunc=80 Size=284 Line=210 Col=1 Alt=0  
  
/*****  
/*                               Status of Devices                               */  
/*****  
  
Devices ,  
    Offline_at_IPL 0000-FFFF,  
    Online_at_IPL  1800-18ff,  
    Online_at_IPL  2000-20ff,  
    Offline_at_IPL 0291-0592,  
    Sensed       0000-FFFF
```

Steps for defining a virtual switch and defining equivalency IDs for OSA devices

This procedure defines the virtual switch for z/VM and how to define equivalency IDs (EQIDs) for your OSA devices. Subsequent topics explain how to set up TCP/IP and your virtual machines to use the virtual switch.

The recommended facility to use when virtual machines need to have network connectivity with each other is a z/VM virtual switch. You can define a virtual switch dynamically using the CP DEFINE VSWITCH command, but attempts by guests to use the virtual switch fail if you do not do this before guests try to use it. Such a situation is disruptive to the Linux boot process. To be sure the virtual switch is always available for your Linux guests, define a virtual switch in the SYSTEM CONFIG file.

For successful guest relocation, you must define a virtual switch with the same name on each SSI member and that virtual switch must use an OSA device

equivalent to the OSA device on the other member(s). In this task, you define equivalency IDs (EQIDs) for those OSA devices.

For an overview on configuring TCP/IP for Linux, see “TCP/IP networking options for Linux” on page 34.

Before you begin: You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to define a virtual switch:

1. Find the section titled “Status of Devices.” At the XEDIT command line, type this command and press the Enter key:

```
====> /status of devices
```

2. Place a slash (/) in the prefix area on the line after the last statement in the “Status of Devices” section. Then, press the Enter key.
3. At the command line, type this command and press the Enter key:

```
====> input
```

4. Type these lines, then press the Enter key.

```
member: Begin
RDEVICE rdev_range EQID eqid TYPE OSA
DEFINE VSWITCH switch_name RDEV rdev [PRIROUTER]
MODIFY VSWITCH switch_name GRANT userid
member: End
```

where:

member

Is the SSI member name; for example, MEMBER1.

rdev_range

Is a range of real device numbers for OSA devices.

eqid

Is an equivalency ID.

switch_name

Is a name you give the virtual switch

rdev

Is the real device address of your QDIO OSA-Express device

userid

Is the user ID of a virtual machine for which you are granting access to the virtual switch.

Note that the PRIROUTER (primary router) option is in brackets, indicating it is optional. The PRIROUTER option is required if any of your Linux virtual servers will act as a router to other networks. If you specify the PRIROUTER option, do not use the brackets.

Example: If the member name is MEMBER1, the device range is 4000-401E, the equivalency ID is OSA1SET3, the virtual switch name is SSISW, the real device address of your OSA-Express device is 4204, and your virtual machine is LINUX0, add these lines:

```
MEMBER1: Begin
  RDEvice 4000-401E EQID OSA1SET3 TYPe OSA
  DEFINE VSWITCH SSISW RDEV 4204
  MODIFY VSWITCH SSISW GRANT LINUX0
MEMBER1: End
```

5. Repeat the MODIFY VSWITCH command with the GRANT option for each user ID you wish to use the virtual switch.
6. Repeat steps 4 on page 64 to 5 for virtual switches on other members.
7. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```
SYSTEM  CONFIG  Z1  F 80  Trunc=80 Size=286 Line=210 Col=1 Alt=0

MEMBER1: Begin
  RDEvice 4000-401E EQID OSA1SET3 TYPe OSA
  DEFINE VSWITCH SSISW RDEV 4204
  MODIFY VSWITCH SSISW GRANT LINUX0
MEMBER1: End
MEMBER2: Begin
  RDEvice 4000-401E EQID OSA1SET3 TYPe OSA
  DEFINE VSWITCH SSISW RDEV 4204
  MODIFY VSWITCH SSISW GRANT LINUX0
MEMBER2: End
MEMBER3: Begin
  RDEvice 4000-401E EQID OSA1SET3 TYPe OSA
  DEFINE VSWITCH SSISW RDEV 4204
  MODIFY VSWITCH SSISW GRANT LINUX0
MEMBER3: End
MEMBER4: Begin
  RDEvice 4000-401E EQID OSA1SET3 TYPe OSA
  DEFINE VSWITCH SSISW RDEV 4204
  MODIFY VSWITCH SSISW GRANT LINUX0
MEMBER4: End
```

Steps for setting addresses for consoles

During the first IPL of your z/VM system, you needed to specify a load parameter so you could communicate with the Stand-Alone Program Loader (SAPL). The reason is the new z/VM system did not know which device address to use to display messages and prompts. The installation system includes default device addresses for use as the system operator console and emergency messages console, but these addresses rarely correspond to your production hardware configuration. So you will not need to use the SAPL each time you IPL z/VM, you need to supply the address of your IPL console and your emergency messages console on the Operator_Consoles statement.

During IPL, CP tries each device on the Operator_Consoles statement (from left to right) until it finds an active device. If no devices on the list are active, CP loads a disabled wait state and terminates.

The emergency message console is used as an additional console during failures. Define the emergency console with the Emergency_Message_Console statement.

Before you begin: You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52. You need to know the real device addresses for the operator console and the emergency message console.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to set up addresses for consoles:

1. Find the Operator_Consoles statement. At the XEDIT command line, type this command and press the Enter key:

```
====> /operator_consoles
```

2. Replace all device addresses on the Operator_Consoles statement with the real device addresses for your operator consoles.

Notes:

- a. You can have one or more console addresses.
 - b. If you are not using 3270 devices on your system and are using 3270 integrated consoles instead, the keyword “System_3270” designates a full-screen integrated 3270 console (also known as SYSG) and the keyword “System_Console” designates an integrated line mode 3270 console (also known as SYSC).
3. Use the same addresses for the Emergency_Message_Console statement as you did for the Operator_Consoles statement.
 4. Save the file. At the XEDIT command line, type this command and press the Enter key:

```
====> save
```

You should now see something like the following:

```
SYSTEM  CONFIG  Z1  F 80  Trunc=80 Size=286 Line=222 Col=1 Alt=0

/*****
/*              Console Definitions              */
/*****

Operator_Consoles      0021 System_3270 System_Console
Emergency_Message_Consoles  0021 System_3270 System_Console
```

Steps for updating special escape character defaults

z/VM provides special escape characters that, when included in terminal input streams, tell CP to perform an action before evaluating subsequent characters in the stream. An important escape character is the line end character; this character, followed by “cp” indicates that what follows is a CP command and should not be given to Linux. The default line end character is “#”.

Example: Issuing:

```
#cp query time
```

sends the QUERY TIME command to CP rather than Linux.

Unfortunately, the default line end character is “#”, which is also a special character for Linux. For example, the prolog to a shell script usually has a character string known as the “shebang” consisting of the characters “#!” followed by the path to the shell program used to evaluate the shell script. If you enter those characters on a virtual machine console (for example, while using the line mode editor ed), CP strips off the # symbol and attempts to evaluate the characters as command input.

You can change the default special escape characters for each virtual machine individually, or set a system-wide default. If the primary purpose of your z/VM system is for Linux virtual servers, it makes sense to change the default system-wide. Chapter 10, “Performing run-time tasks,” on page 119 shows you how to change the special escape characters for an individual virtual machine.

Before you begin: You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

If not already doing so, you must edit the SYSTEM CONFIG file. Type this command and press the Enter key:

```
xedit system config z
```

Perform these steps to update system-wide special escape character defaults:

1. Find the section entitled “Special characters for system set here.” At the XEDIT command line, type this command and press the Enter key:

```
====> /special characters for system set here
```

2. Move the cursor to the line with the text “Escape”, then move it to the first quote mark.
3. Type “OFF”.

```
Char_Delete  OFF      ,      /* System default ... @  */
Escape       OFF      ,      /* System default ... "  */
Line_Delete  OFF      ,      /* Default is cent sign */
```

4. Move the cursor to “Line_End”, and replace “#” with “%”.

```
Line_End     '%'      ,      /* System default ... #  */
```

5. Save the file. At the XEDIT command line, type this command and press the Enter key:

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

You should now see something like the following:

```
SYSTEM  CONFIG  Z1  F 80  Trunc=80 Size=286 Line=233 Col=1 Alt=0

/*****
/*          Special characters for system set here          */
*****/

Character_Defaults ,
Char_Delete  OFF ,      /* System default ... @ */
Escape       OFF ,      /* System default ... " */
Line_Delete  OFF ,      /* Default is cent sign */
Line_End     '%' ,      /* System default ... # */
Tab          OFF ,      /* System default ... " */
```

Steps for checking the syntax of the SYSTEM CONFIG file

Be sure the SYSTEM CONFIG file has the correct syntax by using the CPSYNTAX command.

Before you begin: You need to access to the parm disk. See “Steps for accessing the parm disk and creating a backup for SYSTEM CONFIG” on page 52.

Perform these steps to check the syntax of the SYSTEM CONFIG file:

1. To gain access to the CPSYNTAX command, access the 193 disk as E. From the command line, type the following command and press the Enter key:

```
access 193 e
Ready;
```

2. If you are still editing SYSTEM CONFIG, exit the XEDIT session. From the XEDIT command line, type this command and press the Enter key:

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

3. Run the CPSYNTAX command. From the command line, type this command and press the Enter key:

```
cpsyntax system config z (lpar lparname
```

where *lparname* is the name of an LPAR for an SSI member.

Example:

```
cpsyntax system config z (lpar lp01
Configuration file processing complete -- no errors encountered.
Ready;
```

4. Check for a zero return code. If you do not get a zero return code, modify the SYSTEM CONFIG file.
5. Repeat step 3 and 4 for each LPAR in the SSI cluster.

You are done when you get a zero return code from CPSYNTAX for all LPARs.

Chapter 4. Configuring the Directory Maintenance Facility

This topic describes how to configure the Directory Maintenance Facility (DirMaint). For an overview of DirMaint and user management, see “Planning for user management” on page 37.

If you need to repeat these tasks, be sure to shut DirMaint down.

Steps for enabling DirMaint

These steps have you enable DirMaint. You need to enable DirMaint only on one member of the SSI cluster. Other SSI members have DirMaint satellite servers that send user directory update requests to the member where DirMaint is running.

Before you begin: You need to have a license for the DirMaint product. For ordering information, see the announcement letter for z/VM or contact your IBM representative.

Perform these steps to enable DirMaint:

1. Log onto MAINT620.
2. Check that MAINT620 has write access to MAINT's minidisk 51D.
 - a. From the command line, type this command and press the Enter key:

```
query accessed
Mode Stat   Files  Vdev  Label/Directory
A     R/W      9     191   MNT191
B     R/W     130    5E5   MNT5E5
C     R/W      59    4CC   MNT4CC
D     R/W     185    51D   MNT51D
E     R/O      83    2CC   MNT2CC
S     R/O     689    190   MNT190
Y/S   R/O    1008    19E   MNT19E
Ready;
```

- b. If 51D is not accessed as R/W, type this command and press the Enter key:

```
link maint 51d 51d mr
Ready;
```

Note: If MAINT620 cannot access 51d in R/W mode, another user has the link as R/W and must change its access to R/O. **Do not use mw** mode for MAINT620's link.

- c. Access 51D. From the command line, type this command and press the Enter key:

```
access 51d d
Ready;
```

3. Enable DirMaint. From the command line, type this command and press the Enter key:

```
service dirm enable
```

Result: The command:

- Sets DirMaint as ENABLED within CP.
- Updates the SYSTEM CONFIG file on the CP CF0 parm disk.
- In an SSI cluster, adds a record into the other members' \$PRODS file to enable. The \$PRODS file holds a list of products that were serviced using the SERVICE EXEC.

Note: If enabling DirMaint after using the limited access directory manager function installed for use with the System Management Application Programming Interface (SMAPI), you must configure DirMaint as described in *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* before DirMaint will continue to function successfully with SMAPI.

Continue to the next steps.

Steps for adding DirMaint maintenance and changing the passwords for the DirMaint service machines

As shipped with z/VM, the DirMaint service machines have default passwords, which you must change.

Before you begin: You must enable DirMaint.

Perform the following steps to change passwords:

1. On each member in the SSI cluster:
 - a. Log on to MAINT620.
 - b. Run PUT2PROD to place any outstanding DirMaint service into production. From the command line, type the following and press the Enter key:

```
put2prod dirm
```

- c. Log off.
2. On one member of the SSI cluster:
 - a. Log on to MAINT620.
 - b. Ensure the 2CC minidisk is accessed R/W as C. From the command line, type the following command and press the Enter key:

```
====> query accessed
Mode Stat   Files Vdev  Label/Directory
A     R/W      9   191  MNT191
B     R/W     130 5E5   MNT5E5
C     R/W      59 2CC   MNT2CC
D     R/W     185 51D   MNT51D
S     R/O     689 190   MNT190
Y/S   R/O    1008 19E   MNT19E
Ready;
```

- c. From the command line type the following command and press the Enter key:

```
xedit user direct c
```

- d. Find the directory entry for 6VMDIR20. On the XEDIT command line, type this command and press the Enter key:

```
====> /USER 6VMDIR20
```

- e. Type your password over the existing password. If you need to insert characters, press the Insert key.

Example:

```
USER 6VMDIR20 MYPASSWD 16M 64M EG
:
```

- f. Likewise, change the default passwords for DIRMAINT, DATAMOVE, DATAMOV2 through DATAMOV4, and DIRMSAT2 through DIRMSAT4.
- g. From the XEDIT command line, issue the following command:

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

3. On each system in the SSI:

- a. Log on to MAINT620.
- b. To gain access to the DIRECTXA utility, access the 550 minidisk as F. From the command line, type the following command and press the Enter key:

```
acc 550 f
Ready;
```

- c. From the command line, type the following command and press the Enter key:

```
directxa user direct c
z/VM USER DIRECTORY CREATION PROGRAM - z/VM version 6 release 2
EOJ DIRECTORY UPDATED AND ON LINE
Ready; T=0.13/0.15 14:33:34
```

- d. Log off.

Continue with the following steps.

Steps for configuring DirMaint

6VMDIR20 is the user ID that does service and maintenance for DirMaint. In keeping with the practice for products maintained by the VMSES/E component of z/VM, the virtual machine user ID is the same as the product identifier. In this procedure, you use 6VMDIR20 to configure DirMaint.

Before you begin: Make sure the DIRMAINT virtual machine is logged off.

You need to log onto the 6VMDIR20 virtual machine.

Perform these steps to perform an initial configuration:

1. Access the 492 minidisk:

```
access 492 e
Ready;
```

2. Issue this command:

```
dir2prod access_new 6vmdir20 dirm
:
:
DMSACP726I 492 e released
DIR2PROD: Normal Termination
Ready;
```

3. Create the USER INPUT file on the DIRMAINT 1DF disk for DirMaint's use. From the command line, type the following commands and press the Enter key:

```
link pmaint 2cc 2cc rr
ENTER READ PASSWORD:
password
Ready;
acc 2cc c
Ready;
copyfile user direct c user input j (olddate
```

where *password* is the read password for PMAINT's 2CC.

Before DirMaint is activated, the current USER DIRECT source file on PMAINT's 2CC matches the online user directory. By copying the current USER DIRECT source file from PMAINT's 2CC to the 6VMDIR20's J-disk, you create a starting point for DirMaint. From the USER DIRECT file, DirMaint processes the statements to create its own internal working structure for the user directory.

Important: After performing this step, do not use the USER DIRECT file on PMAINT's 2CC. To remind you that you must not use the file, you could rename it to USER NO-USE or USERDRCT UPBYDIRM.

4. Create a file called CONFIGAA DATADVH L. From the command line, type this command and press the Enter key:

```
xedit configaa datadvh l
```

Tip: Files with the naming convention CONFIG nn DATADVH (nn can be alphabetic or numeric) provide DirMaint override instructions. If there are multiple CONFIG* DATADVH files, all are processed in reverse EBCDIC order (CONFIG99 before CONFIG0 before CONFIGZ9 before CONFIGA, before CONFIG). For now, keep it simple by creating only one, CONFIGAA DATADVH.

5. At the XEDIT command line, type this command and press the Enter key:

```
====> input
```

6. Add the following statements to the file:

```
CONFIGAA DATADVH L2 V 80 Trunc=80 Size=5 Line=1 Col=1 Alt=0
====>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
0 * * * Top of File * * *
1 DISK_CLEANUP= YES
2 ONLINE= IMMED
3 RUNMODE= OPERATIONAL
* * * End of File * * *
```

7. To exit input mode, press the Enter key twice.
8. Save the file. From the command line, type this command and press the Enter key:

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

9. Release the 41F disk. From the command line, type this command and press the Enter key:

```
rel 1  
Ready;
```

Remain logged onto 6VMDIR20. Continue with the next steps.

Steps for authorizing users to perform DirMaint tasks

The AUTHFOR CONTROL file specifies user IDs allowed to perform authorized DirMaint tasks. It would be convenient to specify certain user IDs to do these tasks; otherwise, you need to log on the DIRMAINT virtual machine each time you want to do them. In this procedure, you authorize MAINT to perform DirMaint tasks.

Before you begin: From the previous steps, you need to remain logged onto the 6VMDIR20 virtual machine.

Perform these steps to create the authorization file:

1. Create and edit the AUTHFOR CONTROL file on the J-disk.

```
xedit authfor control j
```

2. On the XEDIT command line, type this command and press the Enter key to enter input mode:

```
====> input
```

3. Type these lines in the input area:

```
:  
ALL MAINT * 140A ADGHOPS  
ALL MAINT * 150A ADGHOPS
```

4. Press the Enter key twice to return to editing mode.
5. From the XEDIT command line, type this command and press the Enter key:

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

Continue to the next steps.

Steps for controlling where DirMaint creates minidisks

The EXTENT CONTROL file tells DirMaint where to allocate minidisks. The file is divided into sections. Each section can occur only once.

:REGIONS.

Identifies DASD extents on individual devices that can be used for minidisk allocation.

:GROUPS.

Allows you to group devices by name and refer to them by a single name. For example, you can create a group called SYSTEM for minidisks related to system user IDs, or a group called LINUX for minidisks related to virtual machines for Linux. Grouping allows you to separate user minidisks from system-related minidisks.

:SSI_VOLUMES.

Can be used if you are operating in an SSI cluster. This section defines which volumes DirMaint uses when cloning a new SUBCONFIG entry from an existing SUBCONFIG entry.

:EXCLUDE.

Gives you the ability to map some or all of a user's extents as excluded. DirMaint prevents existing minidisks from being overlapped during allocation procedures. If you would like to disable this facility for a particular minidisk, specify it within the :EXCLUDE. section. When determining free space, DirMaint does not take into account the extents of excluded devices.

Note: If the minidisk address in the entry is less than 4 digits, pad the address with one or more leading zeros to create a 4-digit address.

:AUTOBLOCK.

Entries are used by the DirMaint machine to calculate the number of cylinders or blocks to allocate with some automatic allocation methods. As shipped, this section is empty. Modifications should be made to this section only if your installation is using a device type not defined in the AUTOBLK DATADVH file.

:DEFAULTS.

Defines the maximum size for various DASD devices. As shipped, this section is empty. Modifications should be made to this section only if your installation is using a device type not defined in the DEFAULTS DATADVH file.

Before you begin: Make sure the DIRMAINT virtual machine is logged off. You need to log onto the 6VMDIR20 virtual machine.

Perform these steps to control where DirMaint creates minidisks:

1. Edit the EXTENT CONTROL file on the J-disk.

```
xedit extent control j
```

2. Find the section labelled ":REGIONS.". From the XEDIT command line, type this command and press the Enter key:

```
====> /:regions.
```

3. Move the cursor to the prefix area for the line containing "*RegionID". Type "i" in the prefix area and press the Enter key:

```
00034 * *****
00035 :REGIONS.
i      *RegionId  VolSer  RegStart  RegEnd  Type
```

4. Move the cursor to the body of the blank line and type a statement with this pattern:

```
RegionId VolSer RegStart RegEnd Type
```


where

RegionId

Is the name of the region. Region names must be unique. If a region entry shares the same name with another region entry, the first record is used, the second entry is ignored. Region names may consist of the characters A-Z, 0-9, #, @, and \$. DirMaint requires that this field be eight characters or less.

VolSer

Is the volume label of the real DASD. This value represents the value placed into volser field on any minidisks generated from this region. Volume IDs supported by DirMaint consist of a maximum of six characters A-Z, 0-9, #, @, and \$. *VolSer* must be unique.

RegStart

Is the starting cylinder for this region.

RegEnd

Is the ending cylinder for this region.

Type

Is the device type of the DASD. You can use default types from DEFAULTS DATADVH file or you may define other types in the :DEFAULTS. section of the EXTENT CONTROL file (see step 11 on page 78.)

Example: The following statement labels an entire 3390-03 with a volume label LX1FA0 as LINUX01:

```
LINUX01 LX1FA0 001 END 3390-03
```

(END is a special keyword specifying the last cylinder.)

5. Repeat step 4 on page 76 for all the volumes you need for your Linux virtual servers.

Note: The region names must be unique.

6. Immediately after the volume statements for Linux virtual servers, add this statement for creating minidisks for other virtual machines.

```
620W02 620W02 1 END devtype
```

where *devtype* is the device type of volume 620W02.

7. Move the cursor to the prefix area for the line containing “*GroupName”. Type “i2” in the prefix area and press the Enter key:

```
00121 :GROUPS.  
i2      *GroupName RegionList
```

8. Type these two lines:

```
GRPLNX (ALLOCATE ROTATING)  
GRPLNX LINUX01
```

If you have additional regions, add those region IDs to the second line.

Example:

```
GRPLNX LINUX01 LINUX02 LINUX03
```

9. Move the cursor to the prefix area for the line with “* USERID ADDRESS”, type “i5” in the prefix area, then press the Enter key:

```
00139 :EXCLUDE.  
i5    * USERID ADDRESS  
00141 :END.
```

10. Type these five lines:

```
MAINT*   012*  
MAINT*   013*  
PMAINT   013*  
SYSDUMP1 012*  
SYSDMP*  012*
```

11. If you have specified any device type that is not listed in the DEFAULTS DATADVH file, define it in the :DEFAULTS. section. Place your cursor into the blank line inside the :DEFAULTS. section, then type a statement with the following pattern:

```
ttttmmm cyl|blk
```

where

```
tttt
```

Specifies the device type associated with this default entry.

```
mmm
```

Specifies an index string used when cross checking a :REGIONS. entry with a :DEFAULTS. entry. The index string should consist of the following characters: A-Z, 0-9, #, @, \$, or – (dash) without embedded blanks. The index string can be from 1- to 4-characters long.

```
cyl/blk
```

Specifies the maximum allocatable cylinder (for CKD devices) or block (for FBA devices) on this device.

Example:

```
3380-05 665  
33801nx1 257
```

Note: If a :REGIONS. entry specifies a *ttttmmm* of 3380-05, DirMaint considers cylinder 664 as the maximum allocatable cylinder.

12. Save the file. From the XEDIT command line, type this command and press the Enter key:

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

Your file should look similar to this:

```

EXTENT  CONTROL  J2  V 254  Trunc=254  Size=89  Line=35  Col=1  Alt=0
====>
35 :REGIONS.
36 *RegionId  VolSer      RegStart      RegEnd  Dev-Type  Comments
:
:
46 LINUX01    430LIN          1          END    3390-03
:
:
57 620W02     620W02          1          END    3390-03
58 :END.
59 :GROUPS.
60 *GroupName RegionList
61 DVHUSR      (ALLOCATE ROTATING)
:
:
65 GRPLNX      (ALLOCATE ROTATING)
66 GRPLNX      LINUX01 LINUX02 LINUX03
:
:
71 :END.
72 :EXCLUDE.
73 * UserId Address
74 MAINT*      012*
75 MAINT*      013*
76 PMAINT      013*
77 SYSDUMP1    012*
78 SYSDMP*     012*
76 :END.
:

```

Continue to the next steps.

Steps for putting the configuration into production and starting DirMaint

In these steps, you place the DirMaint configuration into production and start DirMaint.

Before you begin: You need to log onto 6VMDIR20. Later, when you log onto DIRMAINT, you need to know the log-on password you defined in “Steps for adding DirMaint maintenance and changing the passwords for the DirMaint service machines” on page 72.

Perform these steps to put the configuration into production and start DirMaint:

1. From the CMS command line, type this command and press the Enter key:

```
dir2prod update_files 6vmdir20 dirm
```

2. Copy CONFIGAA DATADVH to the 11F minidisk.

- a. Access the 11F minidisk as F. From the command line, type the following command and press the Enter key:

```
access 11f f
Ready;
```

- b. Copy the file. From the command line, type the following command and press the Enter key:

```
copy configaa datadvh 1 = = f
Ready;
```

3. Log off 6VMDIR20. Type this command and press the Enter key:

```
logoff
```

4. Log onto DIRMAINT.
5. Start the DirMaint program:

```
dvhbegin
:
DVHWAI2140I Waiting for work on 04/02/09 at 18:47:38.
```

6. Log onto MAINT620.
7. Start up the DirMaint satellite server machines. From the command line, type the following and press the Enter key:

```
at membername cmd xautolog dirmsatn
```

where

membername

Is an SSI member name (for example, MEMBER2).

*dirmsat*n**

Is the satellite on that member (for example, DIRMSAT2 on MEMBER2, DIRMSAT3 on MEMBER3, or DIRMSAT4 on MEMBER4).

Note: Do not log on a satellite on the member that runs the DIRMAINT server machine. In these examples, DIRMAINT is on MEMBER1, so you would not log on DIRMSAT.

Repeat the command for the other satellites.

8. Log off MAINT620.

Continue to the next steps.

Steps for testing DirMaint

This procedure tests DirMaint. The command you use allows MAINT to issue DirMaint commands without a password.

Note: You can enter DirMaint commands on any member of the SSI cluster and the command will update all the cluster members.

Before you begin: You need to know the password for MAINT.

Perform these steps to test DirMaint:

1. Log onto MAINT.
2. From the CMS command line, type this command and press the Enter key:

```
dirm needpass no
```

3. When prompted, type MAINT's password and press the Enter key.

You see these messages:

```
dirm needpass no
DVHXT1181R Enter the current logon password of MAINT at DVHTEST6 for
DVHXT1181R authentication. It will not be displayed on the
DVHXT1181R terminal. To exit without processing the command, just
DVHXT1181R press ENTER.

DVHXT1191I Your NEEDPASS request has been sent for processing.
MAINT AT DVHTEST6; T=0.05/0.06 18:53:02
DVHREQ2288I Your USEROPTN request for MAINT at * has been accepted.
DVHBIU3450I The source for directory entry MAINT has been updated.
DVHBIU3456I Object directory update is not required for this source
DVHBIU3456I update.
DVHREQ2289I Your USEROPTN request for MAINT at * has completed;
DVHREQ2289I with RC = 0.
```

You are done.

Step for modifying the OPERATOR's directory entry

In Chapter 9, "Setting up basic system automation," on page 107, you set up the programmable operator facility, which requires that the OPERATOR virtual machine automatically load (IPL) CMS when it is logged on. This procedure uses DirMaint to modify the OPERATOR's directory entry to include a statement that loads CMS.

Before you begin: You need to be logged onto MAINT.

Perform this step to modify the OPERATOR's directory entry:

- From the command line, type this command and press the Enter key:

```
dirm for operator ipl cms
DVHREQ2288I Your IPL request for OPERATOR at * has been accepted.
DVHBIU3450I The source for directory entry OPERATOR has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry OPERATOR have been placed
DVHBIU3428I online.
DVHREQ2289I Your IPL request for OPERATOR at * has completed; with RC
DVHREQ2289I = 0.
```

You know you are done when you see a zero return code from DirMaint.

How to enter DirMaint commands

For z/VM to recognize DirMaint commands, you must type **dirm** or **dirmaint** before them.

Notes:

1. When entering commands from the console of the DIRMAINT, and DATAMOVE service machines, you must omit the keyword **dirm** and **dirmaint**.
2. You can use the short form of the command keyword.

Example: The minimum truncation for the ADD command is "A"; thus, there are three variations for the ADD command:

```
dirm add
dirm ad
dirm a
```

The short form of a command is indicated by capital letters in the syntax diagram for the command. You can find the short form for a DirMaint command in *z/VM: Directory Maintenance Facility Commands Reference*.

Example: To query the OPTION user directory statement of the user LINUX1, you would enter the following:

```
dirm for linux1 option ?
```

where

dirm

Indicates a DirMaint command.

for linux1

The prefix FOR specifies the user ID (in this case, LINUX1) whose directory entry is to be changed or queried.

option

The DirMaint command to be entered. (In this case it is the OPTION command, which targets the OPTION statement in the directory entry for LINUX1.)

? The command parameter. (In this case it is a question mark, which requests that the current OPTION statement be returned in a message.)

For basic DirMaint commands you use, see “Step for managing users” on page 126.

Chapter 5. Configuring TCP/IP

In this topic, you configure TCP/IP on each SSI member, if you have not already done so, and configure TCP/IP to start automatically whenever z/VM is initialized.

For an overview on configuring TCP/IP for Linux, see “TCP/IP networking options for Linux” on page 34.

Setting up the production TCP/IP

During z/VM installation, you had the option to set up a static connection to the TCP/IP network through the IPWIZARD. Such a configuration is suitable for installations that employ only static (as opposed to dynamic) network routes. If you want a more comprehensive TCP/IP network, you need to follow the instructions in *z/VM: TCP/IP Planning and Customization*, SC24-6238.

Table 5 outlines the tasks you must do:

Table 5. Task roadmap for the production TCP/IP

Subtask	Associated instructions (see . . .)
Setting up your production TCP/IP	<p>If you set up a static connection during z/VM installation and are satisfied with this configuration, this task is optional.</p> <p>If you want a more comprehensive TCP/IP network, you need to follow the instructions in <i>z/VM: TCP/IP Planning and Customization</i>, SC24-6238.</p> <p>Note: You must install TCP/IP on each member in the SSI cluster.</p>
Setting up TCP/IP to be automatically started	“Steps for automatically starting TCP/IP”

Steps for automatically starting TCP/IP

This procedure ensures the TCPIP virtual machine is always started when you IPL z/VM. Except for a few system user IDs, the AUTOLOG1 virtual machine automatically starts virtual machines at IPL time through its PROFILE EXEC. By placing an additional XAUTOLOG command for TCPIP in AUTOLOG1's PROFILE EXEC, that virtual machine is automatically logged on when you IPL z/VM.

Before you begin: You need to log on as MAINT on one SSI member.

Perform these steps to ensure TCP/IP is started at IPL time:

1. Link to the AUTOLOG1 191 minidisk. Type this command and press the Enter key:

```
link autolog1 191 091 mr
Ready;
```

2. Access the 091 minidisk as Z. Type this command and press the Enter key:

```
access 091 z
Ready;
```

3. Edit AUTOLOG1's PROFILE EXEC.

- a. Type this command and press the Enter key:

```
xedit profile exec z
```

- b. Go to the line with "PIPE CP XAUTOLOG DTCVSW2", and then type **i** in the prefix area.

```
PROFILE EXEC      A1 F 80 Trunc=80 Size=nnn Line=0 Col=1 Alt=0

00000 * * * Top of File * * *
00001 /*****
00002 /*  AUTOLOG1 PROFILE EXEC                               */
00003 /*****
00004 Address Command
00005 rrc = 0
00006 "PIPE CP XAUTOLOG VMSERVS"
00007 "PIPE CP XAUTOLOG VMSERVU"
00008 "PIPE CP XAUTOLOG VMSERVR"
00009 "PIPE CP XAUTOLOG VMSERVP"
00010 "PIPE CP XAUTOLOG DTCVSW1"
i0011 "PIPE CP XAUTOLOG DTCVSW2"
```

4. Add the new XAUTOLOG statement.

```
"PIPE CP XAUTOLOG TCPIP"
```

5. Press the Enter key.

Result: The file should look like this:

```
PROFILE EXEC      A1 F 80 Trunc=80 Size=nnn Line=0 Col=1 Alt=0

00000 * * * Top of File * * *
00001 /*****
00002 /*  AUTOLOG1 PROFILE EXEC                               */
00003 /*****
00004 Address Command
00005 rrc = 0
00006 "PIPE CP XAUTOLOG VMSERVS"
00007 "PIPE CP XAUTOLOG VMSERVU"
00008 "PIPE CP XAUTOLOG VMSERVR"
00009 "PIPE CP XAUTOLOG VMSERVP"
00010 "PIPE CP XAUTOLOG DTCVSW1"
00011 "PIPE CP XAUTOLOG DTCVSW2"
00012 "PIPE CP XAUTOLOG TCPIP"
:
```

6. Save the file. At the XEDIT command line, type this command and press the Enter key:

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


7. Detach the 091 minidisk (AUTOLOG1's 191). Type this command and press the Enter key:

```
release z (det  
Ready;
```

8. Repeat steps 1 on page 83 to 7 for each SSI member in the cluster.

You are done.

Chapter 6. Restarting and checking each z/VM member in the SSI cluster

After making changes to the system, configuring DirMaint, and setting up the network configuration, restart each z/VM member in the SSI cluster and check that the changes you want were made. This topic explains how to do that.

Steps for restarting each z/VM member

In this procedure, you restart (IPL) each z/VM member, and then log onto MAINT on one member, from which you can check that all your configuration changes are in effect.

Before you begin: You need to be logged on as MAINT on one member (for example, MEMBER1).

Be sure you have checked the syntax of the SYSTEM CONFIG file (see “Steps for checking the syntax of the SYSTEM CONFIG file” on page 68).

Be sure important processing on all members is complete.

Perform these steps to restart each z/VM member:

1. Type the following command and press the Enter key:

```
cp at sysname cmd shutdown reipl
```

where *sysname* is the member name of another SSI member.

2. If you have other SSI members, repeat step 1 for each of them.
3. Shut down the member you are logged onto. Type the following command and press the Enter key:

```
cp shutdown reipl
```

Result: When the system shuts down and re-IPLs, you see a number of IPL messages. z/VM restores the system to the same state as it was prior to shutdown (for instance, with OPERATOR disconnected).

4. Log onto MAINT.

You are done. Go on to the next procedure.

Steps for checking paging, spooling, and CP-owned volumes

Before you begin: You need to be logged on as MAINT.

Perform these steps to check paging, spooling, and CP-owned volumes:

1. Check the paging volumes on each SSI member. For each member, type the following command and press the Enter key:

```

at sysname cmd query alloc page
      EXTENT EXTENT TOTAL  PAGES  HIGH  %
VOLID  RDEV  START   END   PAGES IN USE  PAGE USED
-----
M01P01 C331      0   3338 601020  13646  31175  2%
-----
SUMMARY                601020  13646      2%
USABLE                 601020  13646      2%
Ready; T=0.01/0.01 10:07:56

```

where *sysname* is the member name of an SSI member.

2. Check the spooling volumes on each SSI member. Type the following command and press the Enter key. (The PIPE command filters out all shared volumes, so the response shows volumes for a specific member.)

```

pipe cp at sysname cmd q alloc spool | nlocate /SHARED/ | console
      EXTENT EXTENT TOTAL  PAGES  HIGH  %
VOLID  RDEV  START   END   PAGES IN USE  PAGE USED
-----
M01S01 6281      1   3338 600840  55853  55938  9%
:
:
SUMMARY                600840  55853      9%
USABLE                 600840  55853      9%
Ready; T=0.01/0.01 20:43:15

```

where *sysname* is the member name of an SSI member.

3. Check for volume ownership. Type the following command and press the Enter key:

```

q cpowned
Slot Vol-ID Rdev Type Status SSIOwner SysOwner
1 M01RES 6280 Own Online and attached MYSSI MEMBER1
2 ----- ----- Reserved -----
3 ----- ----- Reserved -----
4 ----- ----- Reserved -----
5 VMCOM1 6284 Own Online and attached MYSSI -----
6 ----- ----- Reserved -----
7 ----- ----- Reserved -----
8 ----- ----- Reserved -----
9 ----- ----- Reserved -----
10 M01S01 6281 Own Online and attached MYSSI MEMBER1
11 M02S01 639C Share Online and attached MYSSI MEMBER2
12 ----- ----- Reserved -----
13 ----- ----- Reserved -----
:
255 M01P01 6282 Own Online and attached MYSSI MEMBER1
Ready; T=0.01/0.01 16:12:11

```

You are done. Go on to the next procedure.

Step for checking the SSI cluster

Before you begin: You need to be logged on as MAINT.

Perform these steps to check the SSI cluster:

- Type the following command and press the Enter key. Check the system IDs and that each member is in the Joined state.

```

q ssi
SSI Name: MYSSI
SSI Mode: Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: VMCOM1 on 6284
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
 1 MEMBER1  Joined      06/20/11    16:12:26 06/20/11    16:12:26
 2 MEMBER2  Joined      06/20/11    16:12:20 06/20/11    16:12:20
 3 MEMBER3  Joined      06/20/11    16:12:28 06/20/11    16:12:28
 4 MEMBER4  Joined      06/20/11    16:12:30 06/20/11    16:12:30
Ready; T=0.01/0.01 16:12:43

```

You are done. Go on to the next procedure.

Step for checking the user volume list

Before you begin: You need to be logged on as MAINT.

Perform this step to check the user volume list:

- Check the user volumes on each SSI member. Type the following command and press the Enter key.

```

pipe cp at sysname cmd q dasd | locate /SYSTEM/ | console
DASD DB35 CP SYSTEM LINUSR1 0
DASD DB36 CP SYSTEM LINUSR2 0
DASD DB3F CP SYSTEM LINUSR3 0
DASD DB40 CP SYSTEM LINUSR4 0
DASD DB40 CP SYSTEM LINUSR5 0
:
Ready; T=0.01/0.01 10:19:29

```

where *sysname* is the member name of an SSI member.

You are done. Go on to the next procedure.

Steps for checking features

Before you begin: You need to be logged on as MAINT.

Note: The system configuration (SYSTEM CONFIG) you modified is common to all SSI members, so you do not need to query each member. However, if you do have different features set on other SSI members through system qualifiers, you need to log on to MAINT on those members to check the features.

Perform these steps to check features:

1. Check the shutdown time periods. Type these commands and press the Enter key after each command:

```

query shutdowntime
System shutdown time: 30 seconds; previous shutdown time: n seconds.
Ready;
query signal shutdowntime
System default shutdown signal timeout: 500 seconds
Ready;

```

2. Check the number of retrieve buffers. Type this command and press the Enter key:

```
query retrieve
20 buffers available. Maximum of 255 buffers may be selected.
Ready;
```

You are done. Go on to the next procedure.

Step for checking offline devices

Before you begin: You need to be logged on as MAINT.

Note: The system configuration (SYSTEM CONFIG) you modified was common to all SSI members, so you do not need to query each member. However, if you do have different offline device settings on other SSI members through system qualifiers, you can use the AT command to query those members for offline devices.

Perform this step to check offline devices:

- Type this command and press the Enter key:

```
query dasd offline
```

You should see devices that you specified as OFFLINE in the SYSTEM CONFIG file.

Example:

```
query dasd offline
DASD 1440 OFFLINE , DASD 1441 OFFLINE , DASD 1442 OFFLINE , DASD 1443 OFFLINE
```

If no DASD volumes are offline, you receive the following response:

```
An offline DASD was not found
```

You are done. Go on to the next procedure.

Step for checking the virtual switches

Before you begin: You need to be logged on as MAINT.

Perform this step to check the virtual switch:

- Check the virtual switches on each SSI member. Type this command and press the Enter key:

```

at sysname cmd query vswitch details
VSWITCH SYSTEM VSWITCH1 Type: QDIO      Connected: 0      Maxconn: INFINITE
PERSISTENT RESTRICTED      NONROUTER      Accounting: OFF
USERBASED
VLAN Unaware
MAC address: 02-24-44-00-00-1A      MAC Protection: OFF
State: Ready
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF
Uplink Port:
RDEV: B4204.P00 VDEV: 4204 Controller: DTCVSW2
EQID: OSA4204
Uplink Port Connection:
MAC address: 00-14-5E-B7-12-FA
RX Packets: 0      Discarded: 0      Errors: 0
TX Packets: 0      Discarded: 0      Errors: 0
RX Bytes: 0      TX Bytes: 0
Device: 4204 Unit: 000      Role: DATA      Port: 2049
Ready; T=0.01/0.01 11:52:14

```

where *sysname* is the member name of an SSI member.

The responses show the virtual switches are defined and ready for use (“State: Ready”), but no virtual machine is using it (“Connected: 0”).

You are done. Go on to the next procedure.

Step for checking character defaults

Before you begin: You need to be logged on as MAINT.

Perform this step to check character defaults:

- Type this command and press the Enter key:

```

query terminal
LINEND % , LINEDEL OFF, CHARDEL OFF, ESCAPE OFF, TABCHAR "
LINESIZE 080, ATTN OFF, APL OFF, TEXT OFF, MODE VM, HILIGHT OFF
CONMODE 3215, BREAKIN IMMED , BRKKEY PA1 , SCRNSAVE OFF
AUTOCR ON , MORE 050 010, HOLD ON , TIMESTAMP OFF, SYS3270 OFF
Ready;

```

The response should show the LINEND character is “%” and that the LINEDEL, CHARDEL, and ESCAPE characters are “OFF.”

You are done. Go on to the next procedure.

Steps for checking DirMaint

Before you begin: You need to be logged on as MAINT.

Perform the following steps to check DirMaint:

1. Check the DIRMAINT and DIRMSAT n server machines in the SSI cluster. Type the following command and press the Enter key:

```

query userid at all

```

where *userid* is DIRMAINT, DIRMSAT, DIRMSAT2, DIRMSAT3, or DIRMSAT4. For example:

```
q dirmsat3 at all
MEMBER3 : DIRMSAT3 - DSC
Ready; T=0.01/0.01 12:36:01
```

2. Similarly, use the QUERY *userid* AT ALL command to check the DATAMOVE and DATAMOV*n* server machines in the SSI cluster.

You are done. Go on to the next procedure.

Steps for checking TCP/IP

Before you begin: You need to log on as TCPMAINT.

Perform these steps to check TCP/IP:

1. From the command line, type this command and press the Enter key:

```
ifconfig -a
name      inet addr: ip_address mask: mask
          UP BROADCAST MULTICAST MTU: 1500
          vdev: vdev rdev: rdev type: QDIO ETHERNET portname: UNASSIGNED portnumb
          transport type: IP
          ipv4 router type: NONROUTER ipv6: DISABLED
          cpu: 0 forwarding: ENABLED ipv4 path MTU discovery: DISABLED
          RX bytes: 448161 TX bytes: 386814
```

You defined the interface name (*name*), IP address (*ip_address*), mask (*mask*), virtual device address (*vdev*), and real device address (*rdev*) for your production TCP/IP through the IPWIZARD (see “Setting up the production TCP/IP” on page 83).

2. To check the network interface, ping the other SSI members and an address outside your subnetwork. Type the following command and press the Enter key:

```
ping ip_address
Ping Level level 620: Pinging host ip_address.
                  Enter 'HX' followed by 'BEGIN' to interrupt.
PING: Ping #1 response took 0.008 seconds. Successes so far 1.
Ready;
```

where *ip_address* is the IP address of another SSI member or an address outside your subnetwork.

You are done checking the system configuration.

Chapter 7. Creating your first Linux virtual machine and installing Linux

This topic covers configuring your first Linux virtual machine and installing your first Linux operating system.

Overview of defining virtual machines for Linux

Previous sections have shown you how to configure z/VM functions and facilities in order to create the system infrastructure for your virtual machines: configuring z/VM, enabling and configuring DirMaint, and configuring TCP/IP. This section and the next, Chapter 8, “Cloning Linux virtual servers,” on page 105, show you how to install your first Linux operating system and then use a replication or cloning process to create additional Linux virtual servers quickly. The cloning process allows you to create a new Linux virtual server without the need to install the Linux operating system from scratch.

Before you begin defining Linux servers, read Chapter 2, “Planning for Linux virtual servers,” on page 27.

The basic subtasks are:

Table 6. Task roadmap for setting up Linux virtual servers

Subtask	Associated instructions (see . . .)
Define a prototype and create your first virtual machine from the prototype	“Steps for defining a master virtual machine for Linux”
Install the Linux operating system in the virtual machine.	Follow the instructions for your Linux distribution. To get started, see “Installing Linux in a virtual machine” on page 99.
Replicate, or clone, additional Linux virtual servers through DirMaint prototype and CLONEDISK functions.	“Steps for cloning a Linux virtual server” on page 105

Steps for defining a master virtual machine for Linux

These steps tell you how to define a master virtual machine that uses the default 2.2 GB space on a 3390-3 DASD.

In this procedure, you modify two sample files, LINDFLT DIRECT and LINUX PROTODIR. LINDFLT DIRECT is a shared profile for all Linux systems and defines common definitions for all your Linux virtual servers. LINUX PROTODIR is designed to define unique characteristics of a virtual machine, such as the DASD definitions.

Before you begin: Log on to MAINT.

Perform these steps to define the master virtual machine:

1. Get the LINDFLT and LINUX samples from DirMaint.

- a. From the command line, type the following commands and press the Enter key:

```
dirm send lindflt direct
dirm send linux protodir
```

Result: DirMaint sends the sample files to MAINT'S reader.

- b. If you have other files in your reader, queue the two files you get from DirMaint with the ORDER command.

Example: DirMaint sent you LINDFLT DIRECT and LINUX PROTODIR with spool identifiers 0004 and 0005, and you already have spool files 0001, 0002, and 0003 in your reader. Issue this command to queue the two files from DirMaint:

```
order rdr 4 5
0000002 FILES ORDERED
Ready;
```

- c. Receive the sample files. From the command line, type these commands and press the Enter key:

```
receive
File LINDFLT DIRECT A1 created from LINDFLT DATADVH A1 received from DIRMAINT at system
receive
File LINUX PROTODIR A1 created from LINUX DATADVH A1 received from DIRMAINT at system
```

2. Edit LINDFLT DIRECT. From the command line, type this command and press the Enter key:

```
xedit lindflt direct a
```

Result: Your LINDFLT DIRECT looks like the following:

```
PROFILE LINDFLT
CLASS G
COMMAND CP TERM LINEND %
COMMAND CP SET PF12 RETRIEVE
COMMAND CP TERM MORE 1 0
COMMAND CP SET RUN ON
COMMAND CP TERM HOLD OFF
IPL 150
MACHINE ESA
MAXSTORAGE 2047M
OPTION CHPIDV ONE
STORAGE 256M
CONSOLE 0009 3215 T
NICDEF 600 TYPE QDIO LAN SYSTEM VSWITCH1
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
```

Notes:

- a. The name of the virtual switch must match the name you used when you defined the virtual switch for the system. See “Steps for defining a virtual switch and defining equivalency IDs for OSA devices” on page 63.

- b. For all virtual machines that you clone using LINDFLT DIRECT, you cannot define other devices at virtual addresses 600, 601, and 602. All those virtual machines will have the same virtual device addresses for the virtual switch.
 - c. For production Linux servers and server virtual machines that perform critical system functions, you can add OPTION QUICKDSP to the directory entry. This option favors the dispatching of the virtual machine over others, thereby assuring good response time for the virtual machine with the option.
3. If you want to give all your cloned Linux virtual machines access to the cryptographic facility:
- a. Add a line before the STORAGE statement:

```
00000 * * * Top of File * * *
00001 PROFILE LINDFLT
00002 CLASS G
00003 COMMAND CP TERM LINEND %
00004 COMMAND CP SET PF12 RETRIEVE
00005 COMMAND CP TERM MORE 1 0
00006 COMMAND CP SET RUN ON
00007 COMMAND CP TERM HOLD OFF
00008 IPL 150
00009 MACHINE ESA
a      MAXSTORAGE 2047M
00011 STORAGE 256M
00012 * * * End of File * * *
```

- b. On the new line, type:

```
CRYPTO APVIRT
```

For more information about the cryptographic facility, see “Giving Linux virtual servers access to cryptographic hardware for SSL acceleration” on page 36.

4. If you make any changes to LINDFLT DIRECT, save the file with the XEDIT FILE command. From the XEDIT command line, type the following command and press the Enter key:

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

5. Edit LINUX PROTODIR.

- a. From the XEDIT command line, type the following command and press the Enter key:

```
xedit linux protodir
```

Result: Your LINUX PROTODIR looks like the following:

```
USER LINUX NOLOG
INCLUDE LINDFLT
MDISK 150 3390 AUTOG 3088 LINGROUP MR
MDISK 151 3390 AUTOG 0200 LINGROUP MR
```

- b. Change each occurrence of “LINGROUP” to “GRPLNX”:

```
MDISK 150 3390 AUTOG 3088 GRPLNX MR
MDISK 151 3390 AUTOG 0200 GRPLNX MR
```

Notes:

- a. For the swap disk 151, you could use a virtual disk, but be aware of the impact a virtual disk has on your system. See “IBM: Performance Considerations for Linux Guests” (www.vm.ibm.com/perf/tips/linuxper.html).
 - b. “GRPLNX” matches the region name you created in “Steps for controlling where DirMaint creates minidisks” on page 75.
6. Save the file. From the XEDIT command line, type the following command and press the Enter key:

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

7. Install the LINDFLT DIRECT and LINUX PROTODIR files.

- a. Start by testing for the existence of a user called LINDFLT. Issue:

```
dirm for lindflt get lock
```

- If the GET LOCK command succeeded, issue:

```
dirm for lindflt replace
```

- If LINDFLT DIRECT is already locked, issue:

```
dirm add lindflt
```

- b. Install LINUX PROTODIR. Issue:

```
dirm file linux protodir
```

8. Use DirMaint to define the master virtual machine (LINMSTR). Type this command and press the Enter key:

```
dirm add linmstr like linux pw new_password
```

where *new_password* is the password for logging on LINMSTR. *new_password* must be 8 characters long and must contain both numbers and letters.

Guidelines:

- You might want to create additional prototypes (*fn* PROTODIR) for meeting differing processing requirements of your Linux virtual servers. You can use naming conventions that indicate the processing requirements satisfied by each prototype. For instance, you might want to create a small memory prototype (128 MB virtual machine called LINUX128), a medium memory prototype (256 MB virtual machine called LINUX256), and a large memory prototype (512 MB virtual machine called LINUX512), then create small, medium, and large master virtual machines, into which you can install the Linux operating system and appropriate application packages or middleware (the large master could have WebSphere, for instance). Then, during the cloning process, you can select the master prototype that meets the needs of your clone.
- The LINUX PROTODIR consumes an entire 3390-3 volume. To help slow the consumption of 3390-3 volumes, you can split the Linux file system into read-only and read/write portions and share the read-only portion. You should be familiar with Linux file system usage practices before attempting to set up read-only portions of the file system. To do this, you need to create a master Linux virtual

server that can update the minidisk to which other Linux virtual servers are linked in read-only mode through the LINK command or LINK directory statement.

Note: There are important maintenance issues here: running Linux virtual servers do not get updates to the read-only disk until those servers are recycled. Also, a running Linux system linked to read-only disks might crash if those disks are updated while it is running. So, you need to develop a maintenance plan to determine how to update the read-only disk and when to recycle Linux virtual servers.

You know you are done when DirMaint tells you LINMSTR is created.

Steps for adding CMS to LINMSTR

It is useful to have CMS available to do z/VM operations, such as using FTP to retrieve files for installing Linux in a virtual machine. However, having access to the CMS minidisks prevents a virtual server from relocating to another member of the SSI cluster. In the previous task, you set up a prototype Linux directory entry without CMS, which allows any clone you create to relocate. For the LINMSTR virtual server (only), you should set up CMS so you can do z/VM operations required to install Linux. The steps in this task tell you how to do that setup.

Before you begin: You must create the LINMSTR directory entry. See “Steps for defining a master virtual machine for Linux” on page 93.

You must log onto MAINT.

Perform the following steps to add CMS to LINMSTR:

1. From the command line, type the following and press the Enter key:

```
dirm for linmstr get
...
RDR FILE nnn SENT FROM DIRMAINT PUN WAS 8116 RECS 0027 CPY 001 A NOHOLD NOKEEP
...
```

where *nnn* is the spool file ID of the reader file.

2. Receive the reader file.

- a. If you have other reader files, queue the new one first.

Example: DirMaint sent you LINMSTR DIRECT with a spool identifier 0004, and you already have spool files 0001, 0002, and 0003 in your reader. Issue the following command to queue the file from DirMaint:

```
order rdr 4
0000001 FILE ORDERED
Ready;
```

- b. From the command line, type the following command and press the Enter key:

```
receive
File LINMSTR DIRECT A1 created from LINMSTR DIRECT A1 received from DIRMAINT at node
```

where *node* is a system name.

3. Edit LINMSTR DIRECT. Type the following command and press the Enter key:

```
xedit linmstr direct
```

4. Place a slash (/) in the prefix area on the line stating “INCLUDE LINDFLT”, and then press the Enter key twice:

```
00000 * * * Top of File * * *
00001 USER LINUX NOLOG
/0002 INCLUDE LINDFLT
00003 MDISK 150 3390 AUTOG 3088 GRPLNX MR
00004 MDISK 151 3390 AUTOG 0200 GRPLNX MR
00005 * * * End of File * * *
```

5. From the XEDIT command line, type the following command and then press the Enter key:

```
====> input
```

6. Type the following statements, and then press the Enter key twice:

```
IPL CMS AUTOCR
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK TCPMAINT 0592 0592 RR
```

7. Save the file. From the XEDIT command line, type the following and press the Enter key:

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

8. From the command line, type the following and press the Enter key:

```
dirm for linmstr replace
PUN FILE 1432 SENT TO DIRMSAT2 RDR AS 1398 RECS 0020 CPY 001 0 NOHOLD NOKEEP
DVHXTM1191I Your REPLACE request has been sent for processing to
DVHXTM1191I DIRMAINT at GDLLCPX1 via DIRMSAT2.
Ready; T=0.01/0.01 13:56:07
DVHREQ2288I Your REPLACE request for LINMSTR at *
DVHREQ2288I has been accepted.
DVHBIU3450I The source for directory entry
DVHBIU3450I LINMSTR has been updated.
DVHBIU3424I The next ONLINE will take place
DVHBIU3424I immediately.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHBIU3428I Changes made to directory entry LINMSTR
DVHBIU3428I have been placed online.
DVHREP3603I Directory entry LINMSTR is now
DVHREP3603I unlocked.
DVHREQ2289I Your REPLACE request for LINMSTR at *
DVHREQ2289I has completed; with RC = 0.
```

You are done when have replaced the LINMSTR directory entry.

Steps for setting up LINMSTR's disks

In this procedure, you create a 192 disk for LINMSTR and access the 592 (TCP/IP) disk. The 192 disk can hold the Linux boot files and this disk will not be replicated in the clones.

Before you begin: You need to log on to MAINT. Later in this procedure, you need to log onto LINMSTR.

Perform these steps to set up LINMSTR's disks:

1. Log onto MAINT and from the command line, type the following command and press the Enter key:

```
dirm for linmstr amdisk 192 devtype autog 50 grplnx label label
DVHXT1191I Your AMDISK request has been sent for processing.
Ready;
DVHREQ2288I Your AMDISK request for LINMSTR at * has been accepted.
DVHSCU3541I Work unit 18124944 has been built and queued for processing.
DVHSHN3541I Processing work unit 18124944 as MAINT from ZVMLINUX,
DVHSHN3541I notifying MAINT at ZVMLINUX, request 62 for LINMSTR sysaffin
DVHSHN3541I *: to: AMDISK 0192 devtype AUTOG 50 GRPLNX
DVHBIU3450I The source for directory entry LINMSTR has been updated.
DVHDRC3428I Changes made to directory entry LINMSTR have just been
DVHDRC3428I placed online.
DVHSHN3430I AMDISK operation for LINMSTR address 0192 has finished (WUCF
DVHSHN3430I 18124944).
DVHREQ2289I Your AMDISK request for LINMSTR at * has completed; with
DVHREQ2289I RC = 0.
```

where *devtype* is the disk device type of the volume (for instance, 3390) and *label* is the disk label.

2. Issue the following command:

```
access 592 z
DMSACP723I Z (592) R/O
Ready; T=0.01/0.01 15:39:45
```

You are done.

Installing Linux in a virtual machine

Though you need to follow the instructions from your Linux distribution to install and configure Linux, and those instructions differ from distribution to distribution, virtual machine facilities and tasks common to all distributions are covered in this section.

Overview of installing Linux in a virtual machine

You must follow your distribution documentation to install the Linux operating system in your new LINMSTR virtual machine, which you created in “Steps for defining a master virtual machine for Linux” on page 93. For information about obtaining your Linux distribution, see “Steps for obtaining documentation and media” on page 48. This topic includes information about your virtual machine configuration that you should know for your Linux distribution.

To get the installation and configuration process started, Linux distributions follow this pattern:

Table 7. Linux installation and configuration process

Stage	Description
1	<p>Use FTP to move the Linux boot files to a target virtual machine. As of this writing, popular Linux distributions require you to ftp the files to your virtual machine.</p> <p>Examples: SuSE SLES uses these boot files:</p> <ul style="list-style-type: none"> • VMRDR IKR. The Linux kernel built for use with a ram disk. • INITRD. The initial ram disk image. • PARM FILE. The generic parm file for use with the ram disk system. <p>Red Hat, Inc., Linux uses these boot files:</p> <ul style="list-style-type: none"> • KERNEL IMG. The Linux kernel. • INITRD IMG. The initial ram disk image. • REDHAT PARM. The generic parm file for use with the ram disk system. <p>Notes:</p> <ol style="list-style-type: none"> 1. You must ftp the Linux kernel and ram disk image files to the virtual machine by specifying them as binary files with a VM record format of fixed length 80. It is recommended that you use the z/VM FTP command to transfer the files. If you use the FTP command, use the commands “bin” and “locsite fix 80” before you make the transfer. If you use the workstation FTP client, issue the commands “bin” and “quote site fixrecfm 80” before you make the transfer. See “Example of using FTP to get the Linux boot files” on page 101. 2. If the hardware management console (HMC) supports accessing HMC removable media from the z/VM logical partition, you do not need a remote FTP server to get the Linux installation files. See “Example of using FTP to install Linux from the hardware management console,” on page 177.
2	<p>Punch the Linux boot files to the virtual machine reader. See “Example of punching Linux boot files to the virtual machine reader” on page 103.</p> <p>Note: Some distributions supply an EXEC that you can use to start the Linux installation (punching the files to the reader, then IPLing the reader). SuSE SLES 10 has an EXEC called <code>sles10.exec</code>, and SLES 11 has an EXEC called <code>sles11.exec</code>, that perform this function. If your distribution has such an EXEC, ftp that EXEC to the virtual machine.</p>
3	<p>Boot (IPL) Linux from the virtual machine reader. See “Example of booting (IPL) the Linux boot files from the virtual machine reader” on page 103.</p>
4	<p>Use the tools and documentation from your Linux distribution to install and configure Linux.</p> <p>Note: Installation of Linux itself, not just the transfer of the installation files, can also be accomplished using FTP.</p>

During Linux configuration, you need to know these important points about the LINMSTR virtual machine configuration:

- When you log on to LINMSTR, CMS is loaded automatically, which allows you to interact with z/VM to get the Linux installation process started.
- Load the Linux boot files onto the **192** disk. You must access 192 as your A-disk:

```
access 192 a
Ready;
```

- The default memory for LINMSTR is 256 MB.
- The 150 minidisk is for the Linux root file system.
- The 151 minidisk is for the Linux swap space.

- LINMSTR connects to the IP network through the virtual network interface connection that you defined in LINDFLT PROTODIR. There are three qeth device addresses that start at the virtual address that you defined on the NICDEF statement.

Example: If you used the statement:

```
NICDEF 600 TYPE QDIO LAN SYSTEM VSWITCH1
```

the qeth device addresses you enter during Linux configuration are 0.0.0600, 0.0.0601, and 0.0.0602.

For information about device drivers, see *Linux on System z: Device Drivers, Features, and Commands* at:

IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

Guidelines:

- Before you install Linux, format the 150 and 151 minidisks. From the CMS command line, type these commands, press the Enter key, then answer the prompts:

```
format 150 z
:
format 151 x
```

- During the Linux installation process, predefine extra device addresses, making it unnecessary to do this after the system has been installed and loaded. When needed, you can add DASD easily through the CP LINK command. This strategy helps you avoid additional steps to add new DASD.
- After you install Linux, check for Linux service packs and apply them before you clone Linux virtual servers.
- Ensure your Linux server /etc/resolv.conf points to the correct name server.
- If you want to use the signal shutdown function for all your Linux systems, you can make the required modification now so it will be included in all clones. See “Steps for enabling Linux virtual servers to shut down automatically” on page 109.
- If you want to monitor the performance of Linux virtual servers with the Performance Toolkit for VM, you might need to install additional software on the master Linux so all clones automatically get the monitoring software. See “Monitoring Linux virtual servers with Performance Toolkit for VM” on page 148.

Example of using FTP to get the Linux boot files

This topic is an example of how to use FTP to get the Linux boot files from a remote server.

Tip: If you do not have an external FTP server or do not want to create an external connection due to security concerns, you can install Linux by accessing a DVD or removable medium in the hardware management console (HMC) through FTP. For more information, see “Example of using FTP to install Linux from the hardware management console,” on page 177.

In this example, parts in bold indicate commands you would issue. First, access as your A-disk a disk large enough to hold the boot files (the 192 disk in the example). Note the command **locsite fix 80**, which sets the VM file format to fixed length 80, the file format necessary for punching the binary files to the virtual machine reader.

Note: Because the directory containing the Linux boot files varies based on the Linux distribution and distribution level, the directory information used in the FTP GET commands in this topic is for example use only.

```
access 192 a
Ready;
ftp 9.185.246.25
VM TCP/IP FTP Level 62
Connecting to 9.185.246.25, port 21
220 linux390 FTP server (Version wu-2.4.2-VR17(1) Thu May 18 03:18:13 EDT 2000)
ready.
USER (identify yourself to the host):
user_ID
>>>USER user_ID
331 Password required for user_ID.
Password:
>>>PASS *****
230 User user_ID logged in.
Command:
get /boot/s390x/sles11.exec
>>>PORT 9,185,246,26,4,8
200 PORT command successful.
>>>RETR /boot/s390x/sles11.exec
150 Opening ASCII mode data connection for sles11.exec (234 bytes).
226 Transfer complete.
234 bytes transferred in 0.091 seconds. Transfer rate 0.49 Kbytes/sec.
Command:
bin
>>>TYPE i
200 Type set to I.
locsite fix 80
Command:
get /boot/s390x/sles11.image
>>>PORT 9,185,246,26,4,5
200 PORT command successful.
>>>RETR /boot/s390x/sles11.image
150 Opening BINARY mode data connection for sles11.image (1511884 bytes).
226 Transfer complete.
1511920 bytes transferred in 1.972 seconds. Transfer rate 766.69 Kbytes/sec.
Command:
get /boot/s390x/initrd sles11.initrd
>>>PORT 9,185,246,26,4,7
200 PORT command successful.
>>>RETR /boot/s390x/sles11.initrd
150 Opening BINARY mode data connection for sles11.initrd (9862505 bytes).
8299680 bytes transferred.
226 Transfer complete.
9862560 bytes transferred in 12.366 seconds. Transfer rate 797.55 Kbytes/sec.
Command:
asc
>>>TYPE a
200 Type set to A.
Command:
get /boot/s390x/parmfile sles11.parmfile
>>>PORT 9,185,246,26,4,8
200 PORT command successful.
>>>RETR /boot/s390x/sles11.parmfile
150 Opening ASCII mode data connection for sles11.parmfile (38 bytes).
226 Transfer complete.
40 bytes transferred in 0.081 seconds. Transfer rate 0.49 Kbytes/sec.
Command:
quit
```

Example of punching Linux boot files to the virtual machine reader

This example shows commands used to punch Linux boot files to the virtual machine reader. When you punch files to the reader, note the spool identifiers (0126, 0127, and 0128 in the example—yours will differ) so if there are other reader files present you can queue (ORDER) the reader files in the proper order.

Note: Some distributions supply an EXEC that you can use to start the Linux installation (punching the files to the reader, then IPLing the reader). SuSE SLES 10 has an EXEC called SLES10.EXEC, and SLES 11 has an EXEC called sles11.exec, that perform this function. If you have such an EXEC, run it instead of punching the files as the example shows.

```
spool punch * close
Ready;
punch vmrdr ikr a (noh
RDR FILE 0126 SENT FROM LINMSTR PUN WAS 0126 RECS 3129 CPY 001 A NOHOLD NOKEEP
Ready;
punch parm file a (noh
RDR FILE 0127 SENT FROM LINMSTR PUN WAS 0127 RECS 3129 CPY 001 A NOHOLD NOKEEP
Ready;
punch suse initrd a (noh
RDR FILE 0128 SENT FROM LINMSTR PUN WAS 0128 RECS 3129 CPY 001 A NOHOLD NOKEEP
Ready;
order reader 126 127 128
0000003 FILES ORDERED
Ready;
```

Tip: If you forget the spool file identifiers, issue QUERY RDR ALL.

Example of booting (IPL) the Linux boot files from the virtual machine reader

This example shows how you get the Linux installation and configuration started by booting (IPL) the virtual machine reader.

```
ipl 00c clear
:
:
Linux version release (root@ikr_rdr.suse.de)
:
:
Command line is: ramdisk_size=32768 root=/dev/ram0 ro
We are running under VM
This machine has no IEEE fpu
Initial ramdisk at: 0x02000000 (9862560 bytes)
Detected device 2000 on subchannel 0000 - PIM = 80, PAM = 80, POM = FF
Detected device 2001 on subchannel 0001 - PIM = 80, PAM = 80, POM = FF
Detected device 0202 on subchannel 0002 - PIM = C0, PAM = C0, POM = FF
Detected device 0192 on subchannel 0003 - PIM = C0, PAM = C0, POM = FF
Detected device 0200 on subchannel 0004 - PIM = C0, PAM = C0, POM = FF
Detected device 0201 on subchannel 0005 - PIM = C0, PAM = C0, POM = FF
Detected device 000C on subchannel 0006 - PIM = 80, PAM = 80, POM = FF
Detected device 000D on subchannel 0007 - PIM = 80, PAM = 80, POM = FF
Detected device 000E on subchannel 0008 - PIM = 80, PAM = 80, POM = FF
:
:
```

Here is an example of some of the errors you might see on your virtual machine console when formatting DASD during the Linux IPL. These errors are normal and are not a problem.

```
dasd_erp(3990): /dev/dasda ( 94: 0),0150001: 074ab5c0: 00000000 00000000 0000
0000 00000000
dasd_erp(3990): /dev/dasda ( 94: 0),0150001: 074ab5d0: 00000000 00000000 0000
0000 00000000
dasd_erp(3990): /dev/dasda ( 94: 0),0150001: Failed CCW (074ab5b8) already lo
gged
```

```
end_request: I/O error, dev 5e:01 (dasd), sector 524320
EXT3-fs error (device dasd(94,1)): read_inode_bitmap: Cannot read inode bitmap -
  block_group = 2, inode_bitmap = 16777472
EXT3-fs error (device dasd(94,1)) in ext3_new_inode: IO failure
EXT3-fs error (device dasd(94,1)): ext3_new_inode: reserved inode or inode > ino
des count - block_group = 0,inode=1
EXT3-fs error (device dasd(94,1)) in ext3_new_inode: IO failure
EXT3-fs error (device dasd(94,1)): ext3_new_inode: reserved inode or inode > ino
des count - block_group = 0,inode=2
EXT3-fs error (device dasd(94,1)) in ext3_new_inode: IO failure
```

At this point, you would continue with instructions for your Linux distribution. See “Overview of defining virtual machines for Linux” on page 93.

Tip: When interacting with the Linux installation program, you might enter virtual machine console mode. For instance, the installation program might tell you to press Enter to accept a default option and pressing the Enter key results in the VM READ state. If this happens, simply press the Enter key again. For more information about virtual machine console states, see “The virtual machine console” on page 7. Also, see “Virtual machine operation tasks” on page 137.

Chapter 8. Cloning Linux virtual servers

This topic covers a rudimentary way to replicate (or clone) Linux virtual servers. In Chapter 7, “Creating your first Linux virtual machine and installing Linux,” on page 93, you created the LINUX prototype and installed the Linux operating system into LINMSTR. In this topic, you create a clone or replica of LINUX, copy the Linux file system from LINMSTR to the clone, and change the IP address of the clone in the Linux configuration files.

Steps for cloning a Linux virtual server

These steps tell you how to clone a Linux virtual server using DirMaint.

Before you begin: You need to

- Be sure that LINMSTR is not running so the new clone can use LINMSTR's IP address. You will use LINMSTR's IP address temporarily for each Linux clone during the cloning process. Since no two Linux virtual servers can have the same IP address at the same time, you must clone Linux servers one at a time, then customize them.
- Have the new IP address for the cloned Linux virtual server.
- Log on as MAINT.

Perform these steps to clone a Linux virtual server:

1. Add a clone of LINUX to the user directory. Issue this DirMaint command:

```
dirm add linux02 like linux pw new_password
```

where you supply a password for *new_password*.

2. Unlock LINMSTR's directory entry. From the command line, type this command and press the Enter key:

```
dirm for linmstr unlock
DVHXT1191I Your UNLOCK request has been sent for processing.
Ready;
DVHREQ2288I Your UNLOCK request for LINMSTR at * has been accepted.
DVHREQ2289I Your UNLOCK request for LINMSTR at * has completed; with
DVHREQ2289I RC = 0.
```

3. Clone the disks from LINMSTR to the clone LINUX02. From the command line, type these DirMaint commands and press the Enter key after each one:

```
dirm for linux02 clonedisk 150 linmstr 150
dirm for linux02 clonedisk 151 linmstr 151
```

Tip: Cloning might take time. You are ready to move to the next step after your work units have completed successfully. To determine the status of your work units, type this command and press the Enter key:

```
dirm query datamove *
```

4. If LINUX02 has not been granted access to a virtual switch, type the following command and press the Enter key:

```
set vswitch switchname grant linux02
```

where *switchname* is the virtual switch name.

Tip: If you grant access, be sure to update the SYSTEM CONFIG file with a new grant for this user ID. See “Steps for defining a virtual switch and defining equivalency IDs for OSA devices” on page 63.

5. Log onto LINUX02.

Result: The Linux operating system loads.

6. Telnet to the new Linux virtual server and update the network configuration files to make permanent the network changes.

Example: You might need to modify the IP address and host name through a tool like yast2.

7. Shut down the Linux operating system in LINUX02 and reboot. Check that the Linux operating system reboots properly and that the network connections are correct.

8. Try pinging this Linux server or try to access a remote site from this Linux server.

You know you are done when the Linux operating system reboots properly and the network connections are correct.

Chapter 9. Setting up basic system automation

This section covers basic forms of z/VM system automation:

- Starting and stopping virtual machines automatically
- Automating z/VM system and virtual machine operations through the programmable operator facility.

Starting and stopping virtual machines automatically

You can start and stop virtual machines manually, but it is more convenient to have z/VM do these tasks automatically. This topic shows you how to:

- Start your Linux virtual servers and important virtual machines automatically. You already configured TCP/IP to start automatically in “Steps for automatically starting TCP/IP” on page 83. Now you will do the same for other virtual machines.

By including the PIPE CP XAUTOLOG command in the AUTOLOG1 virtual machine's PROFILE EXEC, a virtual machine starts automatically when z/VM is loaded (IPL). The AUTOLOG1 user ID is the default user that z/VM logs on at IPL time. When logged on, AUTOLOG1 executes its PROFILE EXEC, which can contain an XAUTOLOG command to start a virtual machine that runs Linux or any other vital system service. Each Linux virtual server directory entry that you clone from the prototype directory contains the statements necessary to load the Linux operating system automatically.

- Stop Linux virtual servers automatically. You can enable Linux to perform an orderly shutdown whenever you shut down z/VM or force the Linux virtual machine to log off with the FORCE command. Because Linux shuts down gracefully, Linux does not have to examine or repair disks when rebooting. Additionally, Linux can notify network monitors that it is shutting down.

Prior to shutting down or forcing a virtual machine to log off, z/VM sends a signal to enabled virtual machines to shut down. The SYSTEM CONFIG file sets the amount of time that z/VM allows a virtual machine operating system to process a shutdown signal. In this topic, you configure Linux to process a shutdown signal.

Related information:

z/VM: CP Planning and Administration, SC24-6178

Steps for automatically starting Linux virtual servers and other virtual machines

This procedure configures the AUTOLOG1 virtual machine to startup virtual machines automatically whenever z/VM starts.

Before you begin: You need to log on as MAINT.

Perform these steps to start virtual machines at startup (IPL) time:

1. Link to the AUTOLOG1 191 minidisk. Type this command and press the Enter key:

```
link autolog1 191 091 mr
Ready;
```

2. Access the 091 minidisk as Z. Type this command and press the Enter key:

```
access 091 z
Ready;
```

3. Edit AUTOLOG1's PROFILE EXEC.

- a. Type this command and press the Enter key:

```
xedit profile exec z
```

- b. Go to the line with "PIPE CP XAUTOLOG TCPIP", and then type / in the prefix area for that line, followed by the Enter key.
- c. From the XEDIT command line, type this command and press the Enter key:

```
====> input
```

4. Add the following XAUTOLOG statement, then press the Enter key:

```
"PIPE CP XAUTOLOG PERFSVM"
```

Note: The statement starts the PERFSVM virtual machine, which you are enabling in anticipation of setting up performance monitoring (see "Using the CP Monitor and Performance Toolkit for VM" on page 153).

5. For each Linux virtual server you want started automatically, add this statement:

```
"PIPE CP XAUTOLOG userid"
```

where *userid* is the user ID of the Linux virtual server.

6. After every third XAUTOLOG statement that starts a Linux virtual server, add this statement:

```
ADDRESS COMMAND CP SLEEP 15 SEC
```

7. Save the file. Press the Enter key to leave input mode, then, at the XEDIT command line, type this command and press the Enter key:

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

8. Detach the 091 minidisk (AUTOLOG1's 191). Type this command and press the Enter key:

```
release z (det
```


9. If you want to test the autolog procedure, from the command line type this command and press the Enter key:

```
xautolog autolog1
```

10. Repeat this task for each member of the SSI cluster.

You are done when the autolog procedure succeeds on all SSI members.

Steps for enabling Linux virtual servers to shut down automatically

Before you begin: You need to have a Linux kernel that supports a shutdown signal, such as SuSE SLES 9 or later. Check your Linux system documentation to determine if it supports the shutdown signal.

You need to log onto MAINT. You also need to log onto your Linux system as a superuser.

Perform these steps to enable Linux virtual servers to shut down automatically:

1. Log on as MAINT on z/VM and check whether your Linux virtual servers are enabled for the shutdown signal. Type the following command and press the Enter key.

Note: Your Linux operating systems need to be running to see if they are enabled for the shutdown signal.

```
q signals
```

Example:

```
q signals
Userid      Signal    Signal Status  Signalled  Timeout
Linux01     SHUTDOWN Enabled        -          -
Linux02     SHUTDOWN Enabled        -          -
Linux03     SHUTDOWN Enabled        -          -
Linux04     SHUTDOWN Enabled        -          -
Linux05     SHUTDOWN Enabled        -          -
Ready;
```

2. While logged on as a superuser on Linux, check the `etc/inittab` file for the following lines:

```
# z/VM or LPAR is shutting down
ca:12345:ctrlaltdel:/sbin/shutdown -h now
```

Important: Be sure the shutdown command shuts down Linux (-h) rather than restarting it (-r).

3. Repeat this task for Linux virtual servers on other members of the SSI cluster.

You are done.

Setting up the programmable operator

This task shows you how to set up a programmable operator on each member in the SSI cluster.

Overview of the programmable operator

The programmable operator facility increases the efficiency of z/VM system operation and allows operation of systems in a distributed processing environment. The facility intercepts all messages and requests directed to the z/VM operator virtual machine and handles them according to programmed actions. Some messages are simply recorded for future reference, others are acted upon programmatically, and others are sent to a real operator to handle.

The programmable operator facility runs in a CMS virtual machine. Although it can run in any virtual machine, because of its programmed capability to log, handle, or redirect messages, it is most commonly run in the CP system operator's virtual machine.

The programmable operator facility compares all messages directed to it against entries listed in a *routing table*, a CMS file. When a match occurs, the prescribed action is performed. If there is no match, no action is performed. Any messages requiring a real operator's response or action are sent to the defined operator at another console, called the *logical operator* console. If the default action is to log messages, all messages are logged in a CMS *log file*. The benefit is that the real system operator sees only important messages, while all messages are recorded for auditing.

Example: As shown in Figure 13, the OPERATOR user ID is defined as the CP system operator when the CP system is configured. Set up the programmable operator to run in the OPERATOR virtual machine and establish another virtual machine with a user ID of LGLOPR. In the routing table, specify LGLOPR as the logical operator. Now any CP or virtual machine user messages sent to the system operator virtual machine can be handled or filtered by the programmable operator or routed to the LGLOPR user ID. A real system operator logs onto LGLOPR and monitors messages sent to that virtual machine.

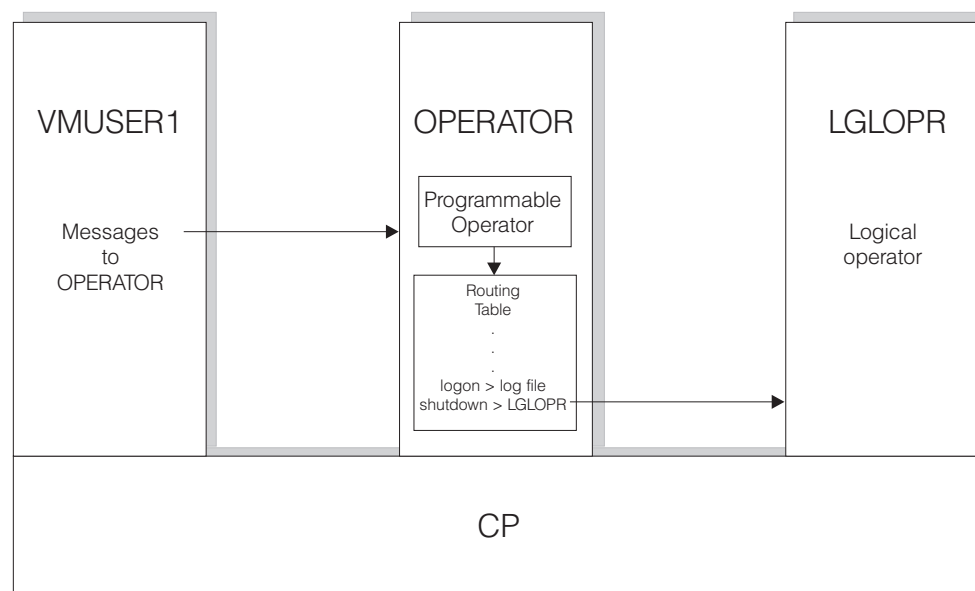


Figure 13. Example of a programmable operator

Related information:

Steps for setting up the routing table

This procedure has you modify the routing table (PROP RTABLE) and add comparison entries for your Linux virtual servers. A comparison entry is a line in the PROP RTABLE that matches a message sent to the programmable operator to an action, such as logging the message or sending the message to the logical operator. For instance, one comparison entry is designed to send a message to the logical operator if a Linux system abnormally terminates. For more information about the statements in the PROP RTABLE, including comparison entries, see “The Routing Table” in *z/VM: CMS Planning and Administration*, SC24-6171.

Through this task, the PROP RTABLE you create is used in common by all OPERATOR virtual machines in the SSI cluster. The task creates a single 692 disk (on which the PROP RTABLE resides) used by all instances of the OPERATOR multiconfiguration virtual machine. Therefore, the PROP RTABLE is common to all OPERATOR instances.

Use this procedure as a model by which to automate other actions for your Linux virtual servers.

Before you begin: You must set up the OPERATOR virtual machine to run CMS (see “Step for modifying the OPERATOR's directory entry” on page 81).

You must log onto MAINT.

Rules: When editing the routing table, you must:

- Keep entries in their proper columns.
- Right-justify numbers in their columns.

Perform these steps to set up the routing table:

1. Type the following command and press the Enter key:

```
dirm for operator amdisk 692 devtype autog 1 grplnx label rr label
DVHXT1191I Your AMDISK request has been sent for processing.
Ready;
DVHREQ2288I Your AMDISK request for OPERATOR at * has been accepted.
DVHSCU3541I Work unit 18124944 has been built and queued for processing.
DVHSHN3541I Processing work unit 18124944 as MAINT from ZVMLINUX,
DVHSHN3541I notifying MAINT at ZVMLINUX, request 62 for OPERATOR sysaffin
DVHSHN3541I *; to: AMDISK 0692 devtype AUTOG 1 GRPLNX
DVHBIU3450I The source for directory entry OPERATOR has been updated.
DVHDRC3428I Changes made to directory entry OPERATOR have just been
DVHDRC3428I placed online.
DVHSHN3430I AMDISK operation for OPERATOR address 0692 has finished (WUCF
DVHSHN3430I 18124944).
DVHREQ2289I Your AMDISK request for OPERATOR at * has completed; with
DVHREQ2289I RC = 0.
```

where *devtype* is the disk device type of the volume (for instance, 3390) and *label* is the disk label.

2. Link to the new OPERATOR 692 disk and access it. Type the following commands and press the Enter key after each one:

```
link operator 692 nnnn mw
Ready;
acc nnnn m
Ready;
```

where *nnnn* is an unused minidisk address.

Example:

```
link operator 692 692 mw
Ready;
acc 692 m
Ready;
```

3. Copy the sample routing table (PROP RTABLE) file from the CMS 190 minidisk to the newly linked disk (for example, the 692 disk). Because the PROP RTABLE file is mode 5 (hidden), you must access the 190 disk as C/A to copy the file. Type these commands and press the Enter key after each one:

```
access 190 c/a
DMSACP723I C (190) R/O
DMSACP725I 190 also = S disk
Ready;
copyfile prop rtable c = = m1
Ready;
rel c
Ready;
```

4. Edit PROP RTABLE M1. Type this command and press the Enter key:

```
xedit prop rtable m
```

- a. To assure that your typing is always uppercase, from the XEDIT command line, type this command and press the Enter key:

```
====> set case u
```

- b. Set line numbers on. From the XEDIT command line, type this command and press the Enter key:

```
====> set num on
```

5. Locate the LGLOPR statement.
- Replace “OPERATOR” with “LGLOPR”.
 - Remove the string “HOSTNODE”.

Result:

```
00001 *      ----- SPECIFY THE PROP CONFIGURATION -----
00002
00003 * IDENTIFY THE LOGICAL OPERATOR
00004
00005 LGLOPR LGLOPR
```

6. Delete these lines (26 to 31). Use the block delete (“dd”) prefix commands in the prefix area:

```
dd      /LOGON                21  26  3
00027 /LOGOFF$~FORCED         21  80  3
00028 /DISCONNECT             21  31  3
00029 /RECONNECT              21  30  3
00030 /DIAL                   21  25  3
dd      /DROP                 21  25  3
```

7. Delete all lines with “PROPNODE” and “NCCF” in them. From the XEDIT command line, type these commands and press the Enter key after each one:

```
====> a11/propnode
```

```
====> delete *
```

```
====> a11/NCCF
```

```
====> delete *
```

```
====> a11
```

8. Save the file. From the XEDIT command line, type this command and press the Enter key:

```
====> save
```

9. Move the current line to the line **before** “SEND ALL OTHER TRAPPED DATA TO THE LOGICAL OPERATOR”. From the XEDIT command line, type these commands and press the Enter key after each one:

```
====> bot
```

```
====> u4
```

```
====> set case mixed
```

“SET CASE MIXED” instructs XEDIT to keep characters you enter in mixed case.

```
====> input
```

10. Add these lines. Remember to keep entries in their proper columns and keep characters in mixed case as shown.

Note: These lines filter many messages. Over time, you might want to remove some if you find you do not use them or want to handle certain messages in other ways, such as logging them.

```
*-----*
* NOTIFY LOGICAL OPERATOR IF LINUX ABENDS
*-----*
/OOPS/                8                DMSPOS  LGLOPR
$DISABLED              8                DMSPOS  LGLOPR
$PSW$                  8                DMSPOS  LGLOPR
$psw$                  8                DMSPOS  LGLOPR
$disconnected/        8                DMSPOS  LGLOPR
/HCP$                  8                DMSPOS  LGLOPR
$FAILURE/              8                DMSPOS  LGLOPR
$failed/               8                DMSPOS  LGLOPR
$Failed/               8                DMSPOS  LGLOPR
$No such/              8                DMSPOS  LGLOPR
$ERROR/                8                DMSPOS  LGLOPR
$error/                8                DMSPOS  LGLOPR
$_MACHINE$             8                DMSPOS  LGLOPR
$cannot open/          8                DMSPOS  LGLOPR
*-----*
* DON'T WORRY ABOUT ANY OTHER SCIF OUTPUT FROM MONITORED USERS
*-----*
                        8
*-----*
```

11. Save the PROP RTABLE file. From the XEDIT command line, type this command and press the Enter key:

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

12. Release and detach the MAINT's linked disk. Type the following command and press the Enter key:

```
rel m (det
```

You are done with the routing table. Go on to the next procedure.

Steps for setting up the programmable operator

Each member of the SSI cluster has an OPERATOR virtual machine that runs the programmable operator. In this task, you configure each OPERATOR virtual machine to start the programmable operator through its PROFILE EXEC, which executes automatically when OPERATOR is logged on. Because each SSI member automatically logs on OPERATOR whenever z/VM initializes, you can have the programmable operator automatically invoked without the need to start it manually.

Before you begin: If OPERATOR is not logged on, you need to log onto the OPERATOR virtual machine. OPERATOR needs to be running CMS (see “Step for modifying the OPERATOR's directory entry” on page 81).

Perform these steps to set up the programmable operator:

1. Log onto OPERATOR and edit its PROFILE EXEC. Type this command and press the Enter key:

```
xedit profile exec
```

2. Go to the bottom of the file and enter input mode. From the XEDIT command line, type these commands and press the Enter key after each one:

```
====> bot
```

```
====> input
```

3. Add the following lines (include the quotes):

```
"ACCESS 692 D"  
"EXEC PROPST PROP DISCONN"
```

4. Press the Enter key twice, then save the file. From the XEDIT command line, type this command and press the Enter key:

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

5. Check that the programmable operator works.

- a. Log onto LGLOPR.
- b. From **OPERATOR**, reload CMS. From the command line, type this command and press the Enter key:

```
ipl cms
```

Result: CMS initializes, executes the PROFILE EXEC, then the OPERATOR virtual machine disconnects. You see initialization statements on LGLOPR's virtual console.

6. Repeat this task on each member of the SSI cluster.

You are done when you see OPERATOR disconnect and the programmable operator initialized.

Important: Once the programmable operator is running, do not log onto OPERATOR except when necessary. If you do log on to OPERATOR, before you can issue commands other than those that control the programmable operator, you must issue the STOP command.

Step for automating Linux virtual consoles

This task shows you how to send Linux virtual console⁷ messages to the programmable operator. Because your Linux virtual servers typically run

7. For a description of virtual consoles, see “Overview of console types” on page 119.

disconnected, you do not see virtual console messages and thus do not know if problems are occurring in the Linux virtual machines. Through this task, you send virtual console messages to the programmable operator for handling. By modifying the routing table, you can record incoming messages or forward important messages to the real operator. Figure 14 shows how automating Linux virtual consoles works.

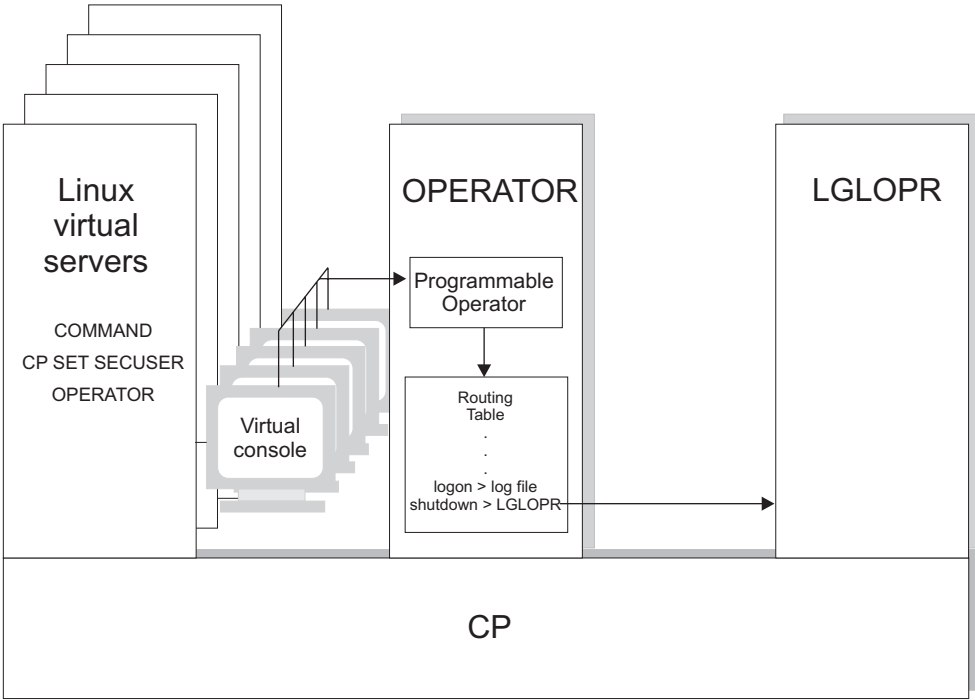


Figure 14. Automating Linux virtual consoles

In the figure, each Linux virtual machine has `COMMAND CP SET SECUSER OPERATOR` in its user directory entry. The command defines the OPERATOR user ID as a *secondary console*, meaning all virtual console messages are sent to OPERATOR. Because OPERATOR is running the programmable operator facility, you can filter messages coming from the Linux virtual machines. You can log the insignificant messages for later reference and forward messages requiring special handling to a real operator at LGLOPR.

By using the secondary console, you can also send messages to the Linux virtual servers to do certain actions, such as attach a network interface (`COUPLE` command), should a Linux failure be detected.

Before you begin: You need to set up the programmable operator. See “Steps for setting up the programmable operator” on page 114.

You must log onto MAINT.

Base your action on the following table:

Table 8. Decision table for automating Linux virtual consoles

If you want to...	Then do the following steps...
Automate all your cloned Linux virtual consoles	<ol style="list-style-type: none"> From the command line, type the following command and press the Enter key: <div>dirm for lindflt get</div> Receive the LINDFLT DIRECT reader file. From the command line, type the following and press the Enter key: <div> <pre>receive File LINDFLT DIRECT A1 created from LINDFLT DATADVH A1 received from DIRMAINT at system</pre> </div> <p>where <i>system</i> is the system name.</p> Edit LINDFLT DIRECT. From the command line, type the following command and press the Enter key: <div>xedit lindflt direct</div> Add a line after the last COMMAND statement. Type a in the prefix area and press the Enter key: <div> <pre>00000 * * * Top of File * * * 00001 PROFILE LINDFLT 00002 CLASS G 00003 COMMAND CP TERM LINEND % 00004 COMMAND CP SET PF12 RETRIEVE 00005 COMMAND CP TERM MORE 1 0 00006 COMMAND CP SET RUN ON a0007 COMMAND CP TERM HOLD OFF 00008 IPL 150 00009 MACHINE ESA 00010 MAXSTORAGE 2047M :</pre> </div> Type the following statement on the new line: <div>COMMAND CP SET SECUSER OPERATOR</div> Save the file. From the XEDIT command line, type the following and press the Enter key: <div>====> file</div> Replace the LINDFLT DIRECT file. From the command line, type the following command and press the Enter key: <div> <pre>dirm for lindflt replace PUN FILE 1456 SENT TO DIRMSAT2 RDR AS 1526 RECS 0027 CPY 001 0 NOHOLD NOKEEP DVHXTM1191I Your REPLACE request has been sent for processing to DVHXTM1191I DIRMAINT at GDLLCPX1 via DIRMSAT2. Ready; T=0.01/0.02 15:52:40 DVHREQ2288I Your REPLACE request for LINDFLT at * DVHREQ2288I has been accepted. : : DVHRLA3891I Your DSATCTL request has been relayed DVHRLA3891I for processing. : : DVHREQ2289I Your REPLACE request for LINDFLT at * DVHREQ2289I has completed; with RC = 0.</pre> </div>

Table 8. Decision table for automating Linux virtual consoles (continued)

If you want to...	Then do the following steps...
Automate the consoles of selected Linux virtual servers	<ol style="list-style-type: none"> From the command line, type the following command and press the Enter key: <div data-bbox="688 340 1239 367" data-label="Text"> <pre>dirm for userid command add cp set secuser operator</pre> </div> Repeat the step for each selected user ID.

where *userid* is the user ID of a Linux virtual server.

Continue to the next procedure.

Steps for testing your automation

To test your automation, try to attempt to couple a network adapter that is already coupled. Messages at the LGLOPR virtual console that the coupling fails tell you that your automation is working properly.

Before you begin: You need to be logged onto LGLOPR and you must have at least one Linux virtual server running.

Perform these steps to test your automation:

- Reset the terminal line end character for LGLOPR. From the command line, type this command and press the Enter key:

```
term linend #
Ready;
```

- From the command line, type this command and press the Enter key:

```
#cp send userid %cp couple 600 system vswitch1
```

where *userid* is the user ID of a Linux virtual server.

Result: You should see these failure messages on the LGLOPR virtual console, which proves your automation is working:

```
Ready;
userid systemid: HCPCPL2788E NIC 0600 not connected; already connected to VS
WITCH SYSTEM
userid systemid: .. VSWITCH1
```

Where *userid* is the user ID of the Linux virtual server and *systemid* is the z/VM system ID.

- Repeat these steps on other members in the SSI cluster.

You know you are done when the test succeeds.

Chapter 10. Performing run-time tasks

This topic covers proactive tasks you need to do to keep z/VM and Linux virtual servers running efficiently. It includes ways for you to monitor z/VM operations so you can prevent system problems.

For monitoring system performance, see Chapter 11, “Monitoring performance and capacity,” on page 147.

Overview of console types

During run-time operations, you can use several consoles. Consoles can be located on the same physical device (workstation, PC, display) or on separate devices. Because of the flexibility of z/VM, the number of consoles varies: you might have a few or many Linux production systems, each with their own console. Regardless of the number of consoles, these are the types of consoles available and a description of each:

- **Hardware management console.** The hardware management console communicates with the hardware. From this console you load (IPL) z/VM itself. You can also do basic z/VM system management from the hardware management console. For details, consult the documentation for your hardware management console.
- **Integrated ASCII console.** The integrated ASCII console can be used to communicate with a Linux guest without the need for any special connectivity or additional hardware, such as control units or network connections. z/VM itself does not exploit the ASCII console, but z/VM can attach it to and detach it from a Linux guest, and query the status of the console. z/VM manages the console as a dedicated device; that is, only one Linux guest can use it at a time. Your Linux guest must be configured to use the ASCII console.

The console provides a VT220 interface for Linux, so you can use Linux tools such as vi and emacs. In an emergency, when no other connectivity is available, this connection could be used, though the console is not limited to such use. For more information, see “Step for managing real devices” on page 122.

- **System operator console.** The system operator console is the console at which CP automatically logs on the primary system operator virtual machine at z/VM load time. In “Steps for setting addresses for consoles” on page 65, you set the real address for this console.

If you do not use the programmable operator facility, you perform system tasks from this console, such as loading z/VM, shutting z/VM down, controlling real devices, communicating with users, and responding to z/VM problems.

- **Logical operator console.** If you followed “Setting up the programmable operator” on page 109, the system console is automated, and you perform system tasks from the logical operator console. In this section, it is assumed you perform real system tasks from the logical operator console.
- **Virtual console for a virtual machine.** When you log onto z/VM as a user, the session you enter is the virtual console. You use this console to control Linux's virtual machine. The tasks you can do from the virtual console include loading Linux, defining the storage size of the virtual machine, and defining virtual CPUs.
- **Linux's system console.** The virtual console becomes the Linux system console once Linux is loaded. Linux system messages are displayed on this console. “Setting up the programmable operator” on page 109 shows you how to send Linux console messages to one central operator.

This section divides up run-time tasks into tasks you can do from the logical operator console (real operation tasks) and tasks you can do from the virtual console (virtual operation tasks). Because the Linux virtual machines have class G command authority and some tasks require the authority of the primary system operator, you must use the logical operator console for those tasks. Other tasks (those for a particular virtual machine running Linux) are executed from the virtual console.

On the logical operator console, you receive messages from the z/VM operating system and from z/VM users (virtual machines). The operating system sends you messages about hardware errors, software errors, and resource shortages. z/VM users who need to have tapes or DASD volumes mounted might send messages to make their requests. Watching the logical operator console lets you respond quickly to system errors and z/VM user requests.

Some tasks you perform from the Linux virtual console are virtual machine tasks; those are documented in this section. Other tasks are operations you perform for the Linux operating system; for those tasks, you must follow the documentation for your Linux distribution.

Real operation tasks

This topic covers the following tasks for operating z/VM itself:

- “Step for monitoring the logical operator console”
- “Step for restarting z/VM” on page 122
- “Step for managing real devices” on page 122
- “Step for managing users” on page 126
- “Step for managing the SSI cluster” on page 133.

Step for monitoring the logical operator console

You will spend much of your time watching each z/VM member in the SSI cluster for changes in system availability and for requests from users for resources.

Before you begin: You must be able to log on as MAINT or onto service virtual machines as indicated.

Perform this step to watch the z/VM logical operator console:

- Base your actions on the following table:

Tip: If you need help understanding the message or knowing what to do, use the Help facility. From the command line, type this command and press the Enter key:

```
help msg message_id
```

where *message_id* is the message identifier.

Example:

```
help msg hcp1nm053e
```

Table 9. Decision table for monitoring the logical operator console

If you receive ...	Then ...
LOGON or LOGOFF messages	Depending on the message, you might need to take action. See “Step for managing users” on page 126.
A paging capacity warning: 90 percent of all paging space is in use.	<ul style="list-style-type: none"> Review system usage and take steps to reduce the system load. Log on as MAINT and add new paging volumes: <ol style="list-style-type: none"> Format and allocate a new paging volume. See “Steps for adding a paging, spooling, temporary disk, or user volume” on page 49. Attach the new paging volume to the system: <div>attach rdev to system</div> <p>where <i>rdev</i> is the real device address.</p> Update the SYSTEM CONFIG file to include the new paging volume. See “Steps for updating the CP-owned volume list” on page 52.
A spooling capacity warning: 90 percent of all spooling space is in use.	<ul style="list-style-type: none"> Review the spool files. You can start additional printers to reduce print spool backlog. If a reader backlog exists, request all users to read in their files. Purge unneeded spool files. You can use SPTAPE to save selected spool files on tape before purging them. Log on as MAINT and add new spooling volumes: <ol style="list-style-type: none"> Format and allocate a new spooling volume. See “Steps for adding a paging, spooling, temporary disk, or user volume” on page 49. Attach the new spooling volume to the system: <div>attach rdev to system</div> <p>where <i>rdev</i> is the real device address.</p> Update the SYSTEM CONFIG file to include the new spooling volume. See “Steps for updating the CP-owned volume list” on page 52.
Device errors	See “Step for managing real devices” on page 122.
Warnings about journaling, erep, and accounting recording: {EREP Accounting SYMPTOM CONFIG} records are accumulating for userid userid.	<ol style="list-style-type: none"> Log onto the user ID indicated in the message. When CMS begins to boot and you see VM READ, type this command and press the Enter key: <div>access (noprof</div> Archive the data on the A-dsik according to your retention policy. Erase the data files on the A-disk. Log off. From LGLOPR, type this command and press the Enter key: <div>xautolog userid</div>

Table 9. Decision table for monitoring the logical operator console (continued)

If you receive ...	Then ...
Complaints about system performance	See Chapter 11, “Monitoring performance and capacity,” on page 147.

You know you are done when the system and users are running efficiently.

Step for restarting z/VM

If you need to restart z/VM, use the SHUTDOWN REIPL command. The SHUTDOWN command sends signals to the Linux virtual servers to shut down and waits for a period of time defined in the SYSTEM CONFIG file.

Before you begin: You must log on as the system operator (if you followed “Setting up the programmable operator” on page 109, that user is LGLOPR).

Perform this step to restart z/VM:

- Type this command and press the Enter key:

```
shutdown reipl
```

Result: After the system shuts down and reboots (IPLs), you see a number of IPL messages. z/VM restores the system to the same state as it was prior to shutdown.

You know you are done when you see z/VM restored.

Step for managing real devices

Before you begin: You must log on as the system operator (if you followed “Setting up the programmable operator” on page 109, that user is LGLOPR).

Perform this step to manage devices:

- Base your actions on the following table:

Tip: If you need more information about a command, use the Help facility. From the command line, type this command and press the Enter key:

```
help cp command_name
```

where *command_name* is the CP command.

Example:

```
help cp vary
```

Table 10. Decision table for managing real devices

If you want to ...	Then use the following command ...
Use the ASCII console to communicate with a Linux virtual machine	<p>CP ATTACH SYSASCII TO <i>userid</i></p> <p>Notes:</p> <ol style="list-style-type: none"> 1. Refer to the Red Hat, Inc., and SUSE Linux configuration tools for more information about the ASCII console support. Also, see the console support section in <i>Linux on System z: Device Drivers, Features, and Commands</i> at: IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html) 2. For Linux to use the ASCII console, the Linux kernel must enable the vt220 scpl and /dev/ttyS1 must be uncommented in the inittab. For SUSE Linux and Red Hat, Inc., the vt220 console is enabled by default, but you must uncomment /dev/ttyS1 in the inittab. <p>Example:</p> <pre>attach sysascii to linux1 SYSASCII attached to LINUX1 Ready;</pre>
Display the status of real devices	<p>CP QUERY <i>rdev1 rdev2 ...</i></p> <p>Examples:</p> <ul style="list-style-type: none"> • For DASD: <pre>query 280 7806 DASD 0280 CP SYSTEM XAUSR1 49 DASD 7806 ATTACHED TO ALAN 7806 R/W 620SPL Ready;</pre> <p>In the first response line, the first number listed is the real device number of the DASD. "SYSTEM" indicates the device is allocated to the system for use as users' minidisks. "XAUSR1" is the volume label of the device. "49" indicates there are 49 active links to the volume.</p> <p>In the second response line, "ATTACHED TO ALAN" indicates the DASD is dedicated to ALAN's virtual machine for its exclusive use.</p> <pre>query db41 14:38:46 DASD DB41 CP OWNED 620RES 79 Ready;</pre> <p>If you see "OWNED", the device is used by the system for paging or spooling activity.</p>

Table 10. Decision table for managing real devices (continued)

If you want to ...	Then use the following command ...
Display the status of real devices (continued)	<ul style="list-style-type: none"> For tape drives: <div> <pre>query 102 TAPE 0102 ATTACHED TO RALPH 0236 R/W Ready;</pre> </div> <p>This means that the magnetic tape drive at the address 0102 is attached in read/write mode to RALPH's virtual machine as the virtual device number 0236.</p> For displays: <div> <pre>query 0bcd GRAF 0BCD ATTACHED TO OLIVER 0010 Ready;</pre> </div> <p>This message means the display device at the real device number 0BCD is attached to OLIVER's virtual machine as the virtual device 0010.</p> For Open Systems Adapter devices: <div> <pre>query 480 OSA 0480 ATTACHED TO LINUX1 0480 Ready;</pre> </div> <p>The first number is the real device number of the Open Systems Adapter device. The last number is the virtual device number which each virtual machine uses to refer to an OSA device.</p>
Display the status of the cryptographic facility	<p>CP QUERY CRYPTO APQS</p> <p>Example:</p> <div> <pre>q crypto apqs AP 02 CEX2C Queue 08 is reserved for dedicated use AP 02 CEX2C Queue 09 is superseded by CEX2A AP 02 CEX2C Queue 10 is superseded by CEX2A AP 02 CEX2C Queue 11 is dedicated to GARDNERK AP 03 CEX2C Queue 08 is superseded by CEX2A AP 03 CEX2C Queue 09 is superseded by CEX2A AP 03 CEX2C Queue 10 is superseded by CEX2A AP 03 CEX2C Queue 11 is dedicated to GARDNERK AP 08 CEX2A Queue 08 is reserved for dedicated use AP 08 CEX2A Queue 09 is installed AP 08 CEX2A Queue 10 is installed AP 08 CEX2A Queue 11 is installed AP 09 CEX2A Queue 08 is installed AP 09 CEX2A Queue 09 is installed AP 09 CEX2A Queue 10 is installed AP 09 CEX2A Queue 11 is installed Ready;</pre> </div>
Enable a logical connection to a real device	<p>CP VARY ONLINE <i>rdev</i></p> <p>Example:</p> <div> <pre>vary online 255 0255 VARIED ONLINE Ready;</pre> </div>

Table 10. Decision table for managing real devices (continued)

If you want to ...	Then use the following command ...
Disable a logical connection to a real device	<p>CP VARY OFFLINE <i>rdev</i></p> <p>Example:</p> <pre>vary offline 255 0255 VARIED OFFLINE Ready;</pre>
Find out channel path status by device	<p>CP QUERY PATHS TO <i>rdev</i></p> <p>Example:</p> <pre>query paths to 0291 Device 0291, Status ONLINE CHPIDs to Device 0291 (PIM) : 19 1B 2E 2F Physically Available (PAM) : + + + + Online (LPM) : + + + + Legend + Yes - No Ready;</pre> <p>In response, CP lists all the channel paths that are physically available to the device and indicates whether the channel paths are logically online or offline.</p>
Enable a logical path between the host system and one or more real devices.	<p>CP VARY ON PATH <i>nn</i> TO <i>rdev</i></p> <p>Example: To make path C2 leading to device 140 available to that device:</p> <pre>vary on path c2 to 140 Vary path c2 online command initiated Vary path c2 online command complete Ready;</pre>
Disable a logical path between the host system and one or more real devices.	<p>CP VARY OFF PATH <i>nn</i> TO <i>rdev</i></p> <p>Example: To make path C2 unavailable to devices 0190 through 0197:</p> <pre>vary off path c2 from 0190-0197 Vary path c2 offline command initiated Path c2 not varied offline from device 0192 because it is the last path to device Vary path c2 offline command complete Ready;</pre>
Display the logical path status and devices with that logical path installed	<p>CP QUERY CHPID <i>nn</i></p> <p>Example: To find out the status of channel path 1A:</p> <pre>query chpid 1a PATH 1A ONLINE TO DEVICES 0190, 0191, 0193 PATH 1A OFFLINE TO DEVICES 0192 Ready;</pre>

Table 10. Decision table for managing real devices (continued)

If you want to ...	Then use the following command ...
Make a channel path available to a device	CP VARY ONLINE CHPID <i>nn</i> Example: <pre>vary online chpid 21 Channel path 21 was successfully varied online Ready;</pre>
Remove a channel path from a device	CP VARY OFFLINE PATH <i>nn</i> FROM <i>rdev</i> Example: <pre>vary offline chpid 21 Channel path 21 was successfully varied offline Ready;</pre>
Connect a real device (for example DASD or OSA-Express) to the virtual machine	ATTACH <i>rdev userid</i> Example: <pre>att 2306 lantzy OSA 2306 ATTACHED TO LANTZY 2306 Ready;</pre>
Add expanded storage to the virtual machine	ATTACH XSTORE <i>userid</i> Issue this command when Linux is shutdown. Example: <pre>attach xstore lantzy 15:49:56 XSTORE MIGRATION STARTED, MAY TAKE SEVERAL MINUTES TO COMPLETE 15:49:56 XSTORE CLEARING STARTED 15:49:57 XSTORE attached, size= 512M Ready;</pre>

You are done when you have successfully managed the real devices.

Step for managing users

Before you begin: You need to log on as the system operator (if you followed “Setting up the programmable operator” on page 109, that user is LGLOPR) or MAINT, as indicated.

Perform this step to manage users:

- Base your actions on the following table:

Tips:

- If you need more information about a CP command, use the Help facility. From the command line, type this command and press the Enter key:

```
help cp command_name
```

where *command_name* is the CP command.

Example:

```
help cp force
```

For CP command information, you can also consult *z/VM: CP Commands and Utilities Reference*.

- For help information for DirMaint commands, use the DIRM HELP command. To display a menu of DirMaint HELP information, enter:

```
dirm help
```

Place the cursor under a command or topic for which you want information and then press the Enter key.

To display information about a specific DirMaint command, type the following command and press the Enter key:

```
dirm help command_name
```

where *command_name* is the DirMaint command.

Example:

```
dirm help add
```

The DirMaint HELP facility recognizes the short form of a DirMaint operand, so `dirm help ad` or `dirm help a` are also valid.

To display information about the DIRMAINT command (DIRMAINT EXEC), type the following and press the Enter key:

```
dirm help dirmaint
```

You can also display information about a message (DVH1093 in this example) by typing the following and pressing the Enter key:

```
dirm help dvh1093
```

For information about DirMaint commands, you can also consult *z/VM: Directory Maintenance Facility Commands Reference*.

Table 11. Decision table for managing users

If you want to ...	Then use the following command ...
Add a new directory entry.	<p>DIRM ADD</p> <p>Issue this command from MAINT.</p> <p>Example: To add a new user with user ID LINUX1, perform the following steps:</p> <ol style="list-style-type: none"> 1. Open a new DIRECT file on your A disk by issuing: <div> <pre>xedit linux1 direct a</pre> </div> 2. Insert the following line into the file: <div> <pre>USER LINUX1 TESTPASS</pre> </div> <p>This statement declares that you are defining a user directory entry with user ID LINUX1 and password TESTPASS.</p> 3. Save the file. At the XEDIT command line, type this command and press the Enter key: <div> <pre>====> file</pre> </div> 4. Add the new entry to the directory by typing the following and pressing the Enter key: <div> <pre>dirm add linux1</pre> </div> <p>You are done.</p> <p>Note: Another useful function of the DirMaint ADD command is the ability to replicate (or clone) virtual machines from an existing prototype. For more information about using templates for cloning virtual machines, see Chapter 8, “Cloning Linux virtual servers,” on page 105.</p>
Delete a virtual machine (remove a directory entry from the source directory)	<p>DIRM PURGE</p> <p>Issue this command from MAINT.</p> <p>Use the Clean option to format the minidisks within the directory entry while command processing.</p> <p>Example: To delete the LINUX1 directory entry and format all the disks within its definition, type the following and press the Enter key:</p> <div> <pre>dirm for linux1 purge clean</pre> </div>

Table 11. Decision table for managing users (continued)

If you want to ...	Then use the following command ...
Retrieve a copy of a user or profile directory entry, introduce new control statements or modify existing ones, then replace the entry.	<p>DIRM GET and DIRM REPLACE</p> <p>Issue this command from MAINT.</p> <p>Example: To add to the empty directory entry LINUX1 the OPTION QUICKDSP statement, perform the following steps:</p> <ol style="list-style-type: none"> 1. Get a copy of the LINUX1 directory entry by typing the following and pressing the Enter key: <div>dirm for linux1 get</div> <p>Result: The GET command automatically locks the directory entry. If you are not planning to replace the entry after GET, then use DIRM GET NOLOCK instead. To remove the lock against the directory entry use the DIRM UNLOCK command.</p> <ol style="list-style-type: none"> 2. Open your reader by typing the following and pressing the Enter key: <div>rdrlist</div> 3. Receive the LINUX1 DIRECT file to your A disk by putting your cursor in front of the LINUX1 DIRECT reader file and pressing PF9. 4. Open the LINUX1 DIRECT A by typing the following and pressing the Enter key: <div>xedit linux1 direct a</div> 5. In the prefix area before the USER statement, type "i" and press the Enter key. 6. In the empty line, type "OPTION QUICKDSP". 7. Save the file. At the XEDIT command line, type this command and press the Enter key: <div>====> file</div> 8. Update the directory entry by issuing the DIRM REPLACE command: <div>dirm for linux1 replace</div> <p>You are done.</p>
Query, alter, or delete the current profile entry for a user. The profile entry consolidates other directory statements that are used in common by multiple directory entries.	<p>DIRM INCLUDE</p> <p>Issue this command from MAINT.</p> <p>The INCLUDE statement specifies the name of a profile entry to be invoked as part of the directory entry.</p> <p>Example: To include the PROFILE LINDFLT into the USER LINUX1, type the following and press the Enter key:</p> <div>dirm for linux1 include lindflt</div>

Table 11. Decision table for managing users (continued)

If you want to ...	Then use the following command ...														
Prevent or allow a user from making any changes to his or her directory entry.	<p>DIRM LOCK/UNLOCK</p> <p>Issue this command from MAINT.</p> <p>The UNLOCK operand is used to remove the lock against the directory entry.</p> <p>Example: To lock the directory entry of the user LINUX1, type the following and press the Enter key:</p> <pre>dirm for linux1 lock</pre>														
Add a minidisk to a Linux virtual server	<p>DIRM AMDISK</p> <p>Issue this command from MAINT.</p> <p>Use the region/volume/group id from your EXTENT CONTROL file to point out which DASD extents to use while allocating a new minidisk. Use AUTOR/AUTOV/AUTOG parameters when allocating within a region/volume/group, respectively.</p> <p>Example: To create a new minidisk for user LINUX1 within the DASD group GRPLNX with virtual address 200, a size of 5 cylinders, and mode of MR, type the following and press the Enter key:</p> <pre>dirm for linux1 amdisk 200 x autog 5 grplnx mr</pre> <p>The following describes the modes:</p> <table> <tr> <td>R</td><td>Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.</td></tr> <tr> <td>RR</td><td>Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.</td></tr> <tr> <td>W</td><td>Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.</td></tr> <tr> <td>WR</td><td>Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.</td></tr> <tr> <td>M</td><td>Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.</td></tr> <tr> <td>MR</td><td>Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.</td></tr> <tr> <td>MW</td><td>Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.</td></tr> </table>	R	Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.	RR	Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.	W	Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.	WR	Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.	M	Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.	MR	Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.	MW	Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.
R	Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.														
RR	Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.														
W	Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.														
WR	Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.														
M	Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.														
MR	Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.														
MW	Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.														

Table 11. Decision table for managing users (continued)

If you want to ...	Then use the following command ...
Change a minidisk size or location.	<p>DIRM CMDISK</p> <p>Issue this command from MAINT.</p> <p>Use the region/volume/group id from your EXTENT CONTROL file to point out which DASD extents to use while allocating a new minidisk. Use AUTOR/AUTOV/AUTOG parameters when allocating within a region/volume/group, respectively.</p> <p>Example: To change the location of minidisk 200 of user LINUX1 to the LINUX01 region and to change its size to 3 cylinders, type the following and press the Enter key:</p> <pre>dirm for linux1 cmdisk 200 x autor 3 linux01</pre>
Delete a minidisk from a directory entry.	<p>DIRM DMDISK</p> <p>Issue this command from MAINT.</p> <p>Use the Clean option to format the minidisk while deleting it.</p> <p>Example: To delete the minidisk from the LINUX1 user definition and format it, type the following and press the Enter key:</p> <pre>dirm for linux1 dmdisk 200 clean</pre>
Request that a LINK control statement be inserted in a directory entry and put into effect automatically each time the user logs on.	<p>DIRM LINK</p> <p>Issue this command from MAINT.</p> <p>Example: To establish an MR link to minidisk 200, which is owned by LINUX1, as minidisk 9200 within a user definition of LINUX2, type the following and press the Enter key:</p> <pre>dirm for linux2 link linux1 200 9200 mr</pre>
Permanently change the line end character for a user (the user's virtual console)	<p>DIRM FOR <i>userid</i> TERM LINEND <i>char</i></p> <p>Issue this command from MAINT.</p> <p>Example:</p> <pre>dirm for linux01 term linend pct</pre>
Permanently change the storage size of a virtual machine	<p>DIRM FOR <i>userid</i> STORAGE <i>nnnM</i></p> <p>Issue this command from MAINT.</p> <p>Example:</p> <pre>dirm for linux01 storage 256M</pre>

Table 11. Decision table for managing users (continued)

If you want to ...	Then use the following command ...
Display the number of logged-on users	<p>CP QUERY USERS</p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <pre>query users 8 USERS, 0 DIALED, 0 NET Ready;</pre>
Display a user ID and device address of the user	<p>CP QUERY <i>userid</i></p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <pre>query linux1 LINUX1 -L008B Ready;</pre>
List logged on users and addresses	<p>CP QUERY NAMES</p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <pre>query names LINUX1 -L008E, LINUX2 -L0094, LINUX3 -L008B, MONWRITE - DSC LINUX4 -L0082, LINUX5 -L003E, LINUX6 -L008A, LINUX6 -L0086 Ready;</pre> <p>“DSC” means the virtual machine is running in disconnected mode.</p>
Grant a user access to a virtual switch	<p>SET VSWITCH</p> <p>Example: To grant LINUX02 access to virtual switch SSISW, issue the following command:</p> <pre>set vswitch ssisw grant linux02</pre>
Log off an active user	<p>CP FORCE <i>userid</i></p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <pre>FORCE LINUX01 GRAF 0020 LOGOFF AS LINUX01 USERS = 875 FORCED BY LGLOPR</pre>
Log on another virtual machine	<p>CP XAUTOLOG</p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <pre>xautolog linux03 Command accepted AUTO LOGON *** LINUX03 USERS = 24 BY LGLOPR Ready;</pre>

Table 11. Decision table for managing users (continued)

If you want to ...	Then use the following command ...																									
Check to see whether a Linux virtual server is enabled for the shutdown signal	<p>QUERY SIGNALS</p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <div><pre>query signals</pre><table><thead><tr><th>Userid</th><th>Signal</th><th>Signal Status</th><th>Signalled By</th><th>Timeout Remaining</th></tr></thead><tbody><tr><td>LINUX01</td><td>SHUTDOWN</td><td>Enabled</td><td>-</td><td>-</td></tr><tr><td>LINUX02</td><td>SHUTDOWN</td><td>Enabled</td><td>-</td><td>-</td></tr><tr><td>LINUX03</td><td>SHUTDOWN</td><td>Enabled</td><td>-</td><td>-</td></tr><tr><td>LINUX04</td><td>SHUTDOWN</td><td>Enabled</td><td>-</td><td>-</td></tr></tbody></table></div>	Userid	Signal	Signal Status	Signalled By	Timeout Remaining	LINUX01	SHUTDOWN	Enabled	-	-	LINUX02	SHUTDOWN	Enabled	-	-	LINUX03	SHUTDOWN	Enabled	-	-	LINUX04	SHUTDOWN	Enabled	-	-
Userid	Signal	Signal Status	Signalled By	Timeout Remaining																						
LINUX01	SHUTDOWN	Enabled	-	-																						
LINUX02	SHUTDOWN	Enabled	-	-																						
LINUX03	SHUTDOWN	Enabled	-	-																						
LINUX04	SHUTDOWN	Enabled	-	-																						
Signal a virtual machine to shut down	<p>CP SIGNAL SHUTDOWN <i>userid</i> WITHIN <i>seconds</i></p> <p>Issue this command from LGLOPR.</p> <p>Example:</p> <div><pre>signal shutdown linux04 within 300 Ready;</pre></div>																									

You are done when you have successfully managed the users.

Step for managing the SSI cluster

By using the AT command, you can manage all members in an SSI cluster from one member. The AT command can be used with most commands other than class G commands. For a complete description of the AT command, see *z/VM: CP Commands and Utilities Reference*.

Base your action on the following table:

Table 12. Decision table for managing the SSI cluster

If you want to...	Then use the following command...
Add an EQID to a real device	<p>SET <i>rdev</i> EQID</p> <p>You must first vary off the device.</p> <p>Example: Issue the following commands:</p> <pre>vary off 4004 4004 varied offline 1 device(s) specified; 1 device(s) successfully varied offline Ready; T=0.01/0.01 12:02:53 set rdev 4004 eqid hello type osa HCPZRP6722I Characteristics of device 4004 were set as requested. 1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created Ready; T=0.01/0.01 12:03:04 vary on 4004 4004 varied online 1 device(s) specified; 1 device(s) successfully varied online Ready; T=0.01/0.01 12:03:11</pre>

Table 12. Decision table for managing the SSI cluster (continued)

If you want to...	Then use the following command...
Remove an EQID from a real device	<p>SET rdev NOEQID</p> <p>You must first vary off the device.</p> <p>Example: Issue the following commands:</p> <pre> vary off 4004 4004 varied offline 1 device(s) specified; 1 device(s) successfully varied offline Ready; T=0.01/0.01 12:03:35 set rdev 4004 noeqid type osa HCPZRP6722I Characteristics of device 4004 were set as requested. 1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created Ready; T=0.01/0.01 12:03:43 vary on 4004 4004 varied online 1 device(s) specified; 1 device(s) successfully varied online Ready; T=0.01/0.01 12:03:49 </pre>
Activate an ISFC link dynamically	<p>ACTIVATE ISLINK <i>link</i></p> <p>You must issue the command from each member.</p> <p>Example: Issue the following commands.</p> <p>On MEMBER1:</p> <pre> act isl 50 12:05:03 Link device 0050 activated. Ready; T=0.01/0.01 12:05:03 </pre> <p>On MEMBER2:</p> <pre> act isl 50 12:05:26 Link device 0050 activated. Ready; T=0.01/0.01 12:05:26 12:05:26 HCPKCL2714I Link device 0050 added to link MEMBER1. </pre> <p>When you issue the command on MEMBER2, MEMBER1 should see:</p> <pre> 12:05:26 HCPKCL2714I Link device 0050 added to link MEMBER2. </pre> <p>Tip: If you add an ISFC link dynamically, ensure that you add a permanent link in the SYSTEM CONFIG file. See “Steps for activating ISFC links between SSI members” on page 56.</p>
Autolog a user on another SSI member	<p>AT <i>member</i> CMD XAUTOLOG <i>userid</i></p> <p>Example: Issue the following command:</p> <pre> at member2 cmd xautolog linux02 Command accepted Ready; T=0.01/0.01 11:58:36 </pre>

Table 12. Decision table for managing the SSI cluster (continued)

If you want to...	Then use the following command...
Grant a user on another member access to a virtual switch	<p>AT <i>member</i> CMD SET VSWITCH <i>vswitch</i> GRANT <i>userid</i></p> <p>Example: Issue the following command:</p> <pre>at member2 cmd set vswitch ssisw grant linux02 Command complete Ready; T=0.01/0.01 11:58:55</pre>
Query where in the SSI cluster a user is logged on	<p>Q <i>userid</i> AT ALL</p> <p>or</p> <p>Q <i>userid</i> AT <i>system</i></p> <p>Example: Issue one of the following commands:</p> <pre>q linux02 at all MEMBER1 : LINUX02 - DSC Ready; T=0.01/0.01 12:00:10</pre> <pre>q linux02 at member1 LINUX02 - DSC Ready; T=0.01/0.01 12:00:50</pre>
Query the status of the SSI cluster	<p>QUERY SSI</p> <p>Before you use QUERY SSI, you must link and access PMAINT's 551 disk:</p> <pre>link pmaint 551 551 rr Ready; acc 551 x DMSACP723I X (551) R/O</pre> <p>Example:</p> <pre>q ssi 12:06:14 SSI Name: MYSSI 12:06:14 SSI Mode: Stable 12:06:14 Cross-System Timeouts: Enabled 12:06:14 SSI Persistent Data Record (PDR) device: VMCOM1 on E00A 12:06:14 SLOT SYSTEMID STATE PDR HEARTBEAT RECEIVED HEARTBEAT 12:06:14 1 MEMBER1 Joined 2011-07-19 12:05:47 2011-07-19 12:05:47 12:06:14 2 MEMBER2 Joined 2011-07-19 12:06:14 2011-07-19 12:06:14 12:06:14 3 MEMBER3 Joined 2011-07-19 12:05:57 2011-07-19 12:05:57 12:06:14 4 MEMBER4 Joined 2011-07-19 12:06:00 2011-07-19 12:06:00 Ready; T=0.01/0.01 12:06:14</pre>

Table 12. Decision table for managing the SSI cluster (continued)

If you want to...	Then use the following command...
Query the status of the SSI cluster (continued)	<p>FORMSSI DISPLAY <i>device</i></p> <p>Before you use FORMSSI, you must link and access PMAINT's 551 disk and the disk that holds the persistent data record (PDR), PMAINT's 141:</p> <pre> link pmaint 551 551 rr Ready; acc 551 x DMSACP723I X (551) R/O link pmaint 141 141 rr Ready; acc 141 v DMSACP723I V (141) R/O </pre>

Example of the FORMSSI command:

```

formssi display 141
HCPPDF6618I Persistent Data Record on device 0141 (label VMCOM1) is for MYSSI
HCPPDF6619I PDR state: Unlocked
HCPPDF6619I time stamp: 07/19/11 16:33:54
HCPPDF6619I cross-system timeouts: Enabled
HCPPDF6619I PDR slot 1 system: MEMBER1
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 07/19/11 16:33:49
HCPPDF6619I last change: MEMBER1
HCPPDF6619I PDR slot 2 system: MEMBER2
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 07/19/11 16:28:16
HCPPDF6619I last change: MEMBER2
HCPPDF6619I PDR slot 3 system: MEMBER3
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 07/19/11 16:33:29
HCPPDF6619I last change: MEMBER3
HCPPDF6619I PDR slot 4 system: MEMBER4
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 07/19/11 16:33:32
HCPPDF6619I last change: MEMBER4
Ready; T=0.01/0.01 16:34:16

```

Table 12. Decision table for managing the SSI cluster (continued)

If you want to...	Then use the following command...
Set a non-functioning member to be in the down state	<p>SET SSI MEMBER <i>member</i> DOWN</p> <p>Example: Before you set a member down, ensure the system is truly unresponsive (for example, the hardware has crashed or the LPAR is in a stopped state) and that the Q SSI output reflects unresponsiveness:</p> <pre>q ssi 16:44:04 SSI Name: MYSSI 16:44:04 SSI Mode: Safe 16:44:04 Cross-System Timeouts: Enabled 16:44:04 SSI Persistent Data Record (PDR) device: VMCOM1 on E00A 16:44:04 SLOT SYSTEMID STATE PDR HEARTBEAT RECEIVED HEARTBEAT 16:44:04 1 MEMBER1 Suspended 2011-07-19 16:43:49 2011-07-19 16:43:49 16:44:05 2 MEMBER2 Unknown 2011-07-19 16:28:16 2011-07-19 16:28:16 16:44:05 3 MEMBER3 Suspended 2011-07-19 16:43:59 2011-07-19 16:43:59 16:44:05 4 MEMBER4 Suspended 2011-07-19 16:44:02 2011-07-19 16:44:02 Ready; T=0.01/0.01 16:44:05</pre> <p>Then issue the following command:</p> <pre>set ssi member member2 down Ready; T=0.01/0.01 16:47:31 q ssi 16:44:04 SSI Name: MYSSI 16:44:04 SSI Mode: Stable 16:44:04 Cross-System Timeouts: Enabled 16:44:04 SSI Persistent Data Record (PDR) device: VMCOM1 on E00A 16:44:04 SLOT SYSTEMID STATE PDR HEARTBEAT RECEIVED HEARTBEAT 16:44:04 1 MEMBER1 Joined 2011-07-19 16:43:49 2011-07-19 16:43:49 16:44:05 2 MEMBER2 Down (declared down) 16:44:05 3 MEMBER3 Joined 2011-07-19 16:43:59 2011-07-19 16:43:59 16:44:05 4 MEMBER4 Joined 2011-07-19 16:44:02 2011-07-19 16:44:02 Ready; T=0.01/0.01 16:48:23</pre>
Relocate a virtual server to another SSI member	See Chapter 13, “Preparing for live guest relocation,” on page 169.

You are done when you successfully manage the SSI cluster.

Virtual machine operation tasks

When you run Linux under z/VM, there are times when you must simulate real hardware functions. Use these CP commands to simulate these real operator functions. Issue CP commands from the Linux virtual console, which is the 3270 session for the virtual machine you log onto.

Tips:

- To use any CP commands, you must precede them with the line end character plus “cp”; if you followed “Steps for updating special escape character defaults” on page 67, that string is “%cp”. As shipped, z/VM uses “#cp”.
- If you are on the virtual console and need to break out of a Linux command, Linux requires a key combination of the logical Not (X'5F') plus “c”, which is equivalent to CNTRL-C. The logical Not symbol is a nonstandard character. Most code pages assign the logical Not symbol to the circumflex key (^). Check your code page for the key that corresponds to the logical Not symbol.

Steps for using CP commands at the Linux virtual console

Before you begin: You must have authority to log on the virtual machine that runs Linux.

Perform these steps to use CP commands at the Linux virtual console:

1. Log on the virtual machine that runs Linux.
2. Base your action on the choices in the table:

Table 13. Decision table for using CP commands at the Linux virtual console

If you want to ...	Then use the following command ...
Change the line end character	<div>TERMINAL LINEND <i>new_char</i></div> <div>Example:</div> <div>%cp terminal linend +</div> <div>Tips:</div> <div><ul style="list-style-type: none">• If you followed “Steps for updating special escape character defaults” on page 67, you defined “%” as a system-wide default line end character. Use this command if you need to change the line end character for the virtual console (3270 session) for an individual Linux server. Note that after changing the line end character, you must use the newly-defined line end character instead of the default.• Some Linux distributions support the hcp utility, which allows you to issue CP commands from the virtual console or telnet session by prefacing the command with “hcp” rather than the line end character plus “cp”.<p>Newer Linux distributions support the vmcp module/command, which allows you to substitute “vmcp” rather than the line end character plus “cp”. Consult documentation for your distribution to determine whether it supports the hcp utility or vmcp.</p></div>
Disconnect your display from z/VM without stopping operations in your virtual machine	<div>DISCONNECT</div> <div>Examples:</div> <div>%cp disconnect DISCONNECT AT <i>hh:mm:ss zone weekday mm/dd/yy</i></div> <div>If you use “vmcp” to address CP:</div> <div>vmcp disconnect DISCONNECT AT <i>hh:mm:ss zone weekday mm/dd/yy</i></div> <div>Tips:</div> <div><ul style="list-style-type: none">• Once you issue DISCONNECT, you remain disconnected until you enter the LOGON command.• When your virtual machine is reconnected using the usual logon procedure, it might be placed in CP console function mode (CP READ). If so, to resume execution on your virtual machine, enter the BEGIN command.</div>

Table 13. Decision table for using CP commands at the Linux virtual console (continued)

If you want to ...	Then use the following command ...
Change the memory size of the virtual machine	<p>DEFINE STORAGE</p> <p>Attention:</p> <ul style="list-style-type: none"> • Issue this command only when Linux is shut down because changing storage sizes resets the virtual machine and any operating system running in it. • The maximum size of your virtual machine is defined in your directory entry. If you need more storage than the maximum, contact your system administrator to increase the maximum size. <p>Examples:</p> <pre>%cp define storage 512m STORAGE = 512M Storage cleared - system reset.</pre>
Define additional virtual CPUs	<p>DEFINE CPU <i>cpuaddr</i></p> <p>Examples:</p> <pre>%cp define cpu 1 CPU 01 DEFINED</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp define cpu 1 CPU 01 DEFINED</pre>
Display the status of the cryptographic facility	<p>Q V CRYPTO</p> <p>Examples:</p> <pre>%cp q v crypto AP 63 CEX2A Queue 01 shared Ready;</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp q v crypto AP 63 CEX2A Queue 01 shared Ready;</pre>

Table 13. Decision table for using CP commands at the Linux virtual console (continued)

If you want to ...	Then use the following command ...
Detach virtual CPUs and devices from the virtual machine	<p>DETACH</p> <p>Examples:</p> <ul style="list-style-type: none"> Virtual CPUs: <p>Attention: Issue this command only when Linux is shut down because detaching virtual CPUs resets the virtual machine and any operating system running in it.</p> <pre>%cp detach cpu 01 00: CPU 01 detached 00: Storage cleared - system reset.</pre> Minidisks: <pre>%cp detach 200 DASD 0200 DETACHED</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp detach 200 DASD 0200 DETACHED</pre> Expanded storage: <pre>%cp detach xstore</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp detach xstore</pre> Guest LAN segments: <pre>%cp detach lan qdio0 LAN VMUSERX QDIO0 is destroyed</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp detach lan qdio0 LAN VMUSERX QDIO0 is destroyed</pre>
Link to another disk (read-only mode)	<p>LINK <i>userid vdev1 vdev2</i> RR</p> <p><i>vdev1</i> is <i>userid</i>'s virtual address; <i>vdev2</i> is an available virtual address in your virtual machine configuration.</p> <p>Examples:</p> <pre>%cp link linux01 150 150 rr</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp link linux01 150 150 rr</pre>

Table 13. Decision table for using CP commands at the Linux virtual console (continued)

If you want to ...	Then use the following command ...
Load (boot) Linux in a virtual machine	<p>IPL</p> <p>IPL (initial program load) and boot are synonymous.</p> <p>Examples:</p> <pre>%cp ipl 150 clear</pre> <p>If Linux is running::</p> <pre>shutdown -r now</pre>
Make CP display the HOLDING status when the display screen is full and highlighted output appears on the screen.	<p>TERMINAL HOLD ON</p> <p>When the display screen is in HOLDING status and you want to clear the display, press the PA1 or CLEAR key.</p> <p>Examples:</p> <pre>%cp terminal hold on</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp terminal hold on</pre>
Change the number of seconds that elapse between the time when CP issues the MORE... state and sounds the terminal alarm, and between the time when CP sounds the alarm and clears the display.	<p>TERMINAL MORE <i>mm nn</i></p> <p>Examples:</p> <pre>%cp terminal more 5 2</pre> <p>If you use “vmcp” to address CP:</p> <pre>vmcp terminal more 5 2</pre> <p>means CP waits 5 seconds after issuing the MORE... state before sounding the terminal alarm, then 2 seconds after sounding the alarm before clearing the display.</p>

You are done.

Archiving and backing up critical data

Archiving stores large bodies of data (for example, an entire disk image) for safekeeping, and should be a part of your disaster recovery plan. The data should be mutually consistent, so you can be running, but cannot be making changes. Do archiving on regular intervals, such as weekly or monthly, or whenever you do major software changes. These archives allow you to restore entire systems quickly.

In contrast to archiving, backups store incremental changes, such as changes to individual files, and you can do backups during system operations. Backups are part of an on-demand file-level recovery system and you should do backups daily.

To restore backups, you need a running system, so after a system disaster, use your archive to restore the entire system, then use your backups to restore files.

Sometimes the terms “archive” and “backup” are used to mean the same thing. This topic uses the term “archive” and describes how to archive large bodies of data such as disks. For backups on z/VM, use an incremental backup system, such as Tivoli® Storage Manager. For incremental backup procedures, consult the documentation for your backup product.

On z/VM, archive:

- SSI cluster-wide volumes
- The z/VM system disks on each SSI member
- z/VM release volumes
- The z/VM spooling system, especially named saved systems (NSSs) and discontinuous saved segments (DCSSs)
- Each major Linux kernel or installed software change.

It is not necessary to archive the z/VM paging space, since these disks hold data valid only for the currently-running z/VM system.

z/VM provides two service programs for archiving:

- The DASD Dump Restore (DDR) utility program allows you to create archives of minidisks and complete DASD volumes. The program does not do incremental backups, so all data on a disk is archived whether or not it has changed. There are two versions of the program: one is the DDR command, which you can issue from CMS; the other is a standalone program that you can load (IPL).
- SPXTAPE produces an archive of spool files. Because NSSs (like CMS) and DCSSs are part of the spooling system, you should archive the spooling system. If problems develop with the spooling system and you need to do a CLEAN start of z/VM, it is much easier to restore archived NSSs and DCSSs instead of rebuilding them.

In an SSI cluster, you need to archive only one copy of the NSSs/DCSSs (or one copy of each NSS/DCSS version, if you have multiple versions). You can use SPXTAPE to restore an NSS/DCSS to each member of the SSI cluster.

Related information:

- “DDR” in *z/VM: CP Commands and Utilities Reference*, SC24-6175
- “SPXTAPE” in *z/VM: CP Commands and Utilities Reference*, SC24-6175

Overview of archiving z/VM system data

At installation time, you should have archived your complete z/VM system, including all of your installation volumes. During the procedure, you created a tape that has the DDRXA program and a copy of all the system disks. DDRXA is a standalone utility program that you can load (IPL) from the tape, then use to restore the system to disk. Because DDRXA does not require an operating system, the archive tape is useful for disasters and disaster tests in which z/VM must be restored.

Also at installation time, you used the SPXTAPE command to archive name saved systems, for example CMS, and discontinuous saved segments. It is also important to archive DCSSs used by Linux virtual servers because DCSSs must be common within the SSI cluster and this is the way to ensure they are identical.

Table 14 directs you to procedures to follow for archiving the z/VM system disks, named saved systems, and discontinuous saved segments.

Table 14. Task roadmap for archiving system data

Subtask	Associated instructions (see . . .)
Archiving named saved systems and discontinuous saved segments	"Back up the named saved systems and segments to tape" in <i>z/VM: Installation Guide</i> , GC24-6246
Archiving the z/VM system disks	"Back up the z/VM system to tape" in <i>z/VM: Installation Guide</i> , GC24-6246

Archiving virtual server disks

You should archive your Linux virtual server system minidisks after you install Linux or make a software change, such as adding an application. By archiving the system minidisks (for instance, the 150 and 151 disks), you have the ability to restore your Linux operating system quickly in case of an emergency.

To archive your virtual server minidisks, use the DDR command in CMS. The DDR command requires CMS and a running z/VM system. If you do not have a running z/VM system, restore it from the loadable tape you created as an archive of z/VM (see "Overview of archiving z/VM system data" on page 142).

Because the data on the minidisks must be mutually consistent, you cannot be changing the minidisk contents while you archive. Archiving is best done while the virtual server is down.

Steps for archiving virtual server disks

Before you begin: Shut down the Linux virtual server. You need to log onto MAINT.

Perform these steps to archive virtual server disks:

1. Mount a scratch tape on a tape drive.
2. Attach the tape to MAINT. Type this command and press the Enter key:

```
attach rdev maint 181
```

where *rdev* is the real device address of the tape drive.

3. Link to the Linux virtual server disk you want to archive. From the command line, type this command and press the Enter key:

```
link userid vdev1 vdev2 rr
```

where *vdev1* is the virtual device address of a Linux minidisk and *vdev2* is an available virtual device address in MAINT's configuration.

Example: To link MAINT to LINUX01's 150 minidisk, issue this command:

```
link linux01 150 950 rr
```

4. From the CMS command line, type the DDR command and press the Enter key:

```
ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
```

DDR prompts for control statements by issuing the ENTER: prompt.

5. At the ENTER: prompt, type this command and press the Enter key:

```
sysprint cons
```

6. Designate the disk from which DDR dumps the data. Type this command and press the Enter key:

```
input vdev2 dasd
```

where *vdev2* is MAINT's virtual device address (corresponding to the Linux virtual server disk).

Example:

```
input 950 dasd
```

7. Designate the tape to which the data is to be dumped. Type this command and press the Enter key:

```
output 181 tape scratch
```

8. Dump the disk to tape. Type this command and press the Enter key, then respond to the prompts as shown:

```
dump all
HCPDDR711D VOLID READ IS LIN150
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
:
DUMPING LIN150
:
ENTER:
```

9. To exit DDR, press the Enter key.
10. Repeat these steps for each virtual server disk you want to archive.

Continue with the next procedure.

Steps for restoring virtual server disks

You should test the restoration procedure so you know what to expect in case you have a system disaster. To test the restoration, use a test Linux virtual server with disk sizes the same as your archived Linux disks. When you are satisfied with the restoration procedure, you can remove the test Linux virtual server.

Before you begin: You need a test Linux virtual server. Follow the steps to clone a virtual server in Chapter 8, “Cloning Linux virtual servers,” on page 105. The test Linux virtual server must be shut down and logged off. You need to log onto MAINT.

Perform these steps to restore virtual server disks:

1. Mount the archive tape on a tape drive.
2. Attach the tape to MAINT. Type this command and press the Enter key:

```
attach rdev maint 181
```

where *rdev* is the real device address of the tape drive.

3. Link to the test Linux virtual server disk to which you want restore the archive. From the command line, type this command and press the Enter key:

```
link userid vdev1 vdev2 w
```

where *vdev1* is the virtual device address of the Linux minidisk and *vdev2* is an available virtual device address in MAINT's configuration.

Example: To link MAINT to LINTTEST's 150 minidisk, issue this command:

```
link lintest 150 950 w
```

4. From the CMS command line, type the DDR command and press the Enter key:

```
ddr  
z/VM DASD DUMP/RESTORE PROGRAM  
ENTER:
```

DDR prompts for control statements by issuing the ENTER: prompt.

5. At the ENTER: prompt, type this command and press the Enter key:

```
sysprint cons
```

6. Designate the tape from which DDR restores the data. Type this command and press the Enter key:

```
input 181 tape
```

7. Designate the disk to which the data is to be restored. Type this command and press the Enter key:

```
output vdev2 dasd
```

where *vdev2* is MAINT's virtual device address (corresponding to the test Linux virtual server disk).

Example:

```
output 950 dasd
```

8. Restore the tape to disk. Type this command and press the Enter key, then respond to the prompts as shown:

```
restore all
HCPDDR711D VOLID READ IS LIN150
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
:
:
RESTORING   LIN150
:
:
ENTER:
```

9. To exit DDR, press the Enter key.
10. Repeat these steps for each virtual server disk you want to restore.
11. Log onto the test Linux virtual server to see if it initializes properly.

You know you are done when the restoration is successful.

Chapter 11. Monitoring performance and capacity

This topic explains how to monitor z/VM functions such as paging and spooling space. The topic includes run-time characteristics and symptoms you should observe to avoid system problems.

Overview of performance monitoring

To be successful in monitoring and tuning your z/VM system, you first need to decide what the phrase “good performance” means for your installation. What is considered good performance varies from one installation to another. For example, for some installations response time for commands is the primary indicator of performance, and good performance might mean sub-second response times. You need to evaluate your own workload, decide what the best indicators of performance are for your workload, and decide what values for those indicators constitute good performance in your environment.

Tuning a z/VM system is really nothing more than matching your hardware's capabilities to your workload's characteristics and to your definition of good performance. Through routine monitoring, you can keep track of the characteristics of your workload, the utilization of your hardware, and, most important, whether your definition of good performance is being met.

As you work at tuning your z/VM system, keep in mind that the changes and adjustments you make serve only to get the best performance from the physical resources you have. Sometimes there is nothing more you can do for performance, due to the limitations of your hardware; when all else fails, you need to consider purchasing more.

Because performance ultimately depends on the hardware available, it's important that you, the performance analyst, have an accurate inventory of the physical resources present on your machine. In particular, as you consider tuning your z/VM system, have an accurate inventory of the hardware available to run your workload:

- How many real CPUs are there, and how fast are they?
- How much real storage is available? How is that real storage distributed between central storage and expanded storage (XSTORE)?
- How well is the system equipped for paging?
 - How much total paging space is there?
 - How many paging volumes are there?
 - How are the paging volumes distributed among disk controllers?
 - How fast are the paging volumes and their disk controllers?
 - How many channel paths (CHPIDs) are there from the processor to each disk controller used in paging?
 - For each volume used for paging, are there other kinds of data on the volume besides paging space?
- How well is the system equipped for spooling? The paging questions apply to the spooling configuration, too.

Tuning is the art of optimizing some performance measure of a workload within the constraints imposed by the hardware available. Because the hardware always imposes some type of constraint, tuning often becomes a balancing act, trading off the needs of the whole system against the needs of specific virtual machines.

Sometimes providing more resources to one virtual machine is detrimental to other virtual machines or to the system as a whole. For instance, you might not want virtual machines used for testing programs to get as much system resources as production virtual machines, because your objective is to give production work maximum throughput. Performance monitoring can help you understand which changes meet your performance objectives.

You can use the INDICATE command (class B, C, and E users) to obtain an informal snapshot of system load, scheduler dispatcher queues, and I/O. The QUERY SHARE command (class A and E) allows you to view the share of system resources a virtual machine has. Both of these commands provide only a snapshot, while the MONITOR command collects large amounts of performance measurement data for later systematic and comprehensive analysis.

The MONITOR command (class A and E users) starts and stops the emission of data relevant to specific system events. The command also starts and stops the collection, and periodic emission, of sample data descriptive of system performance.

Monitor *domains* divide the emitted data into topic areas: behavior of I/O, behavior of the scheduler, configuration of the z/VM system, settings of the monitor itself, and so on. Data the monitor emits under a certain domain includes both event information (such as a user logoff) and sample information (collected performance measurement data, usually counters).

Monitoring Linux virtual servers with Performance Toolkit for VM

z/VM provides analysis tools, such as Performance Toolkit for VM, that helps you analyze the data you collect with the MONITOR command. In addition to analyzing z/VM performance data, Performance Toolkit for VM processes Linux performance data, provided you have the proper support software. To process Linux performance data, you have these choices:

- Install a commercial Linux on a system that contains a mainframe performance monitoring product.
- Use a Linux 2.6 kernel such as SLES11, which has built-in support that allows Performance Toolkit for VM to monitor Linux virtual servers.
- For additional performance data from Linux, install RMF™ Performance Monitor and configure Performance Toolkit for VM to access this data.

Note: The RMF Performance Monitor is produced by an IBM Technical Study and is not a part of any IBM product.

Related information:

- For more about z/VM performance, see *z/VM: Performance*, SC24-6208.
- For more about Performance Toolkit for VM, see *z/VM: Performance Toolkit Guide*, SC24-6209 and *z/VM: Performance Toolkit Reference*, SC24-6210.

Overview of the z/VM scheduler

This topic gives a rudimentary introduction to the z/VM scheduler so you can understand system responses to commands such as INDICATE.

The z/VM scheduler controls the dispatching of virtual machines by managing three scheduling lists:

- The *dormant list* contains a list of virtual machines that are idle or waiting for completion of a long event, such as a tape read.
- The *eligible list* contains a list of virtual machines waiting for resources. Each virtual machine is classified according to its processing characteristics (known as its *transaction class*). A scheduler *transaction* is the amount of time the virtual machine remains able to run. A virtual machine that runs short transactions consumes little processor resource between waits, while one that runs long transactions consumes a large amount of processor resource between waits. The transaction classes are:
 - E0. Virtual machines in this class do not wait in the eligible list, but move to the dispatch list immediately.
 - E1. Virtual machines in this class are those doing short transactions, such as interactive users.
 - E2. Virtual machines in this class do medium-length transactions.
 - E3. Virtual machines in this class do long transactions.

The scheduler assesses each virtual machine for its need for available resources. For the scheduler, *resources* include processors, central storage, and paging capacity. When a virtual machine is waiting for resources, it is waiting for the z/VM scheduler to decide that there is enough processor capacity, storage capacity, and paging capacity to add this virtual machine to the set of virtual machines on the dispatch list.

The scheduler calculates the eligible priority of a virtual machine when it enters the eligible list. This priority is called the *deadline priority*, the time by which the virtual machine should enter the dispatch list. The relative priorities of virtual machines are designed to slow down virtual machines that require highly-demanded resources, grant virtual machines their shares of available system resources, and control the amount of resource each class gets. For instance, though E2 and E3 virtual machines wait longer in the eligible list, they receive longer elapsed time slices in the dispatch list, which allows efficient use of system resources and the rapid completion of interactive work.

- The *dispatch list* contains a list of virtual machines that can run or whose wait state is expected to be short (for instance, waiting for a page-in). When a virtual machine enters the dispatch list, it retains its transaction class; E0 virtual machines become Q0, E1 become Q1, and so forth. Also, each virtual machine gets a *deadline priority*, the time of day when the virtual machine should complete its next dispatch time slice under ideal conditions. The lower the dispatch priority, the closer the virtual machine is to being dispatched.

Because dispatching priorities are dynamically calculated, the sequence of the virtual machines in the dispatch list varies according to the changing operating characteristics of the virtual machines.

The scheduler controls the cycling of virtual machines through the three lists. When a virtual machine logs on, it is placed in the dormant list and moved to the eligible list only when it has work to do. When entering the eligible list, the virtual machine is assigned its deadline priority based on its share, resource requirement, and contention for system resources. When resources become available, the scheduler moves virtual machines from the eligible list to the dispatch list and assigns them dispatch priorities. As virtual machines consume CPU time in the dispatch list, they are examined and reassigned priority as their dispatch time slices end. Because a virtual machine consumes a given amount of processing or storage resource, becomes idle, or is preempted in favor of other virtual machines in the eligible list, it moves back to the eligible list (if it can still be dispatched) or the dormant list (if it can no longer be dispatched).

Related information:

For a complete description of the scheduler, see “Virtual Machine Scheduling and Dispatching” in *z/VM: Performance*, SC24-6208.

Steps for taking a snapshot of system performance

Before you begin: You need class E authority. You need to understand scheduler basics and the scheduler lists. See “Overview of performance monitoring” on page 147.

Perform these steps to take a snapshot of system performance:

1. Log on a user ID with class E authority.
2. Base your action on the following table.

Note: You should monitor all systems in the SSI cluster. You can view performance data for other members in the SSI cluster by using the AT command. The AT command has the following pattern:

`at member cmd command`

where *member* is the name of an SSI cluster member and *command* is a command. For example:

`at member2 cmd indicate load`

Table 15. Decision table for system performance

If you want to check ...	Then issue the following command ...
System load	cp indicate load

Example:

```
cp indicate load
AVGPROC-061% 03
XSTORE-000001/SEC MIGRATE-0000/SEC
MDC READS-000028/SEC WRITES-000001/SEC HIT RATIO-100%
STORAGE-015% PAGING-0006/SEC STEAL-000%
Q0-00002(00000)                                DORMANT-00487
Q1-00001(00000)                                E1-00000(00000)
Q2-00001(00000) EXPAN-002 E2-00000(00000)
Q3-00009(00000) EXPAN-002 E3-00000(00000)
PROC 0000-057%                                PROC 0001-060%
PROC 0002-067%
LIMITED-00000
```

Notes:

1. AVGPROC indicates utilization of all CPUs.
2. MIGRATE=0 means all working sets of the logged-on virtual machines fit in central storage and expanded storage. A growing MIGRATE number indicates that central storage and expanded storage are insufficient to contain the working sets of the logged-on virtual machines, so the system has to resort to paging to DASD. The higher the MIGRATE number is, the greater is your workload's paging demands on the amount of real storage (central plus expanded) on your system.
3. STORAGE indicates the real storage utilization.
4. Q0 — Q3 are virtual machines in the dispatch list.
5. E1 — E3 are virtual machines in the eligible list.
6. DORMANT indicates virtual machines that are dormant.

Table 15. Decision table for system performance (continued)

If you want to check ...	Then issue the following command ...
Virtual machines in the dispatch and eligible lists	cp indicate queues exp Example: <div> <pre> cp indicate queues exp TCP/IP Q0 PS 00006309/00005795 -101.6 A02 BKW Q1 R00 00001339/00001352 .I.. -91.50 A00 RSCS Q0 PG 00000614/00000613 -1.227 A02 TKNAIRB Q3 R 00016601/00016504 ..D. .0084 A02 REFSNID Q3 R01 00001708/000015420601 A01 FARMAN Q3 IO 00003528/00003368 ..D. .2457 A02 EDLLNX2 Q3 PS 00017257/00017144 99999 A02 VTAM Q0 PS 00001865/00001864 .I.. 99999 A02 SSLSERV Q3 PS 00000374/00000345 99999 A02 VMLINUX1 Q3 PS 00000987/00000987 ..D. 99999 A02 VMLINUX Q3 PS 00002854/00002854 99999 A02 EDLLNX1 Q3 PS 00005165/00005165 99999 A02 CORAK2 Q3 PS 00147973/00147973 99999 A02 </pre> </div>
	Notes: <ol style="list-style-type: none"> Virtual machines are listed in priority order. When INDICATE QUEUES EXP consistently shows E1, E2, or E3 (eligible) users, an eligible list is forming because one or more system resources appears to be constrained. The third column explains why a virtual machine is in a wait state: <ul style="list-style-type: none"> <i>Rnn</i> — current RUNUSER on the specified real CPU, where <i>nn</i> is the CPU ID (in hexadecimal) IO — waiting for I/O PS — PSW wait (enabled wait state) PG — waiting for paging
I/O	cp indicate i/o (several times to see a pattern) Example: <div> <pre> cp indicate i/o FCONX 73DC TCP/IP310 ---- VMBACKUP ---- TCP/IP ---- MONWRITE 7389 RTMTST 73DF TOMDEF 80C9 </pre> </div>
	Notes: <ol style="list-style-type: none"> INDICATE I/O shows virtual machines currently in an I/O wait state and also shows the real device number to which the most recent I/O operation was mapped. Four dashes (----) indicates a virtual device. If you see the same real device number for several virtual machines, this might be an indication that there are many minidisks on the same DASD or you have a DASD controller bottleneck, where too many DASD doing I/O are on the same controller.

Table 15. Decision table for system performance (continued)

If you want to check ...	Then issue the following command ...
Whether a virtual machine is hung	<p>cp indicate user <i>userid</i> exp (several times at 10-second intervals)</p> <p>Example:</p> <pre> cp indicate user bkW exp Userid=BKW Mach=XC V=V Attached xstore=NONE Iplsys=CMS Devnum=27 Spool: Reads=335 Writes=129 Owned spaces: Number=1 Owned size=247M Primary space: ID=BKW:BASE PRIVATE Defined size=256M Address limit=814M Private spaces: Number=1 Owned size=247M Pages: Main=1369 Xstore=2 Dasd=0 Locked=0 WS=1349 Reserved=0 Shared spaces: Number=0 Owned size=0 Pages: Main=0 Xstore=0 Dasd=0 Locked=0 Private paging: Xstore: Reads=8 Writes=10 Migrates=0 Dasd: Reads=0 Writes= 0 Shared paging: Xstore: Reads=0 Writes= 0 Migrates=0 Dasd: Reads=0 Writes= 0 CPU 00: Ctime=0 07:36:37 Vtime=0 00:00:02 Ttime=0 00:00:03 Rdr=404 Prt=0 Pch=16 IO=3970 </pre>
<p>Notes:</p> <ol style="list-style-type: none"> 1. If you do not see VTIME, TTIME, and IO increasing, the virtual machine might be in a legitimate enabled wait state, but this also indicates the virtual machine might need to be restarted. Warn users and contact the virtual machine owner before you restart the virtual machine. 	
A virtual machine's share of resources	<p>cp query share <i>userid</i></p> <p>Example: Assume your installation has four active virtual machines.</p> <pre> cp query share linux0 LINUX0 : ABSOLUTE SHARE = 40% MAXIMUM SHARE = NOLIMIT cp query share linux1 LINUX1 : RELATIVE SHARE = 200 MAXIMUM SHARE = NOLIMIT cp query share linux2 LINUX2 : RELATIVE SHARE = 100 MAXIMUM SHARE = NOLIMIT cp query share linux3 LINUX3 : RELATIVE SHARE = 100 MAXIMUM SHARE = NOLIMIT </pre>
<p>Notes:</p> <ol style="list-style-type: none"> 1. A virtual machine receives its proportion of any scarce resource (CPUs, real storage, or paging I/O capability) according to its SHARE setting. 2. LINUX0 has an absolute 40% share and therefore receives up to approximately 40% of available resources. 3. LINUX1 receives up to 200/400 (LINUX1's relative share divided by the total relative shares) of the remaining 60% of available resources. 4. LINUX2 and LINUX3 each receive up to 100/400 of the remaining 60% of available resources. 	

Table 15. Decision table for system performance (continued)

If you want to check ...	Then issue the following command ...
--------------------------	--------------------------------------

Paging **cp query alloc page**

Example:

```
cp query alloc page
      EXTENT      EXTENT  TOTAL  PAGES  HIGH  %
VOLID RDEV   START    END   PAGES  IN USE  PAGE USED
-----
K4E40A C621      1      3338 600840  1846  1898  1%
K41006 1006      1      3338 600840  1335  1375  1%
K41007 1007      1      3338 600840  1839  1890  1%
K4PAG1 DB3B      1      3338 600840  2442  2462  1%
K4PAG2 DB3C      1      3338 600840  2209  2247  1%
K4PAG3 DB3D      1      3338 600840  2340  2394  1%
K4PAG4 DB3E      1      3338 600840  2621  2621  1%
K4TDSK DA06      0         0    180    180 100%
-----
SUMMARY              4107K 14812      1%
USABLE              4107K 14812      1%
Ready;
```

Spooling **cp query alloc spool**

Example:

```
cp query alloc spool
      EXTENT      EXTENT  TOTAL  PAGES  HIGH  %
VOLID RDEV   START    END   PAGES  IN USE  PAGE USED
-----
K4551C 8B2E      1      3338 600840 599500 600840 99%
K4551D 8B2F      502     3338 510660 508033 510660 99%
K4961C 961C      501     2654 323100 321442 323100 99%
K4E504 C021      1      3338 600840  43651  43651   7% DUMP
K47808 7808      0      3338 601020  81642 600988 13%
K4951D 951D      0      3338 601020 588935 601020 97%
-----
SUMMARY              3162K 2093K      66%
USABLE              2575K 2050K      79%
Ready;
```

You are done.

Using the CP Monitor and Performance Toolkit for VM

This topic has:

- An overview of the CP Monitor and Performance Toolkit for VM
- Configuring Performance Toolkit for VM
- Using Performance Toolkit for VM to analyze performance and capacity.

Overview of the CP Monitor and Performance Toolkit for VM

The CP Monitor collects system performance data. This data can be processed by an external data reduction program to produce formatted reports or real-time displays that give you an understanding of system operation or help you analyze the use of, and contention for, major system resources. These resources include CPUs, storage, I/O devices, and the paging subsystem. You can control the amount and nature of the data collected. As Figure 15 on page 154 shows, monitoring is in this order:

Table 16. How monitor data is collected

Stage	Description
1	You use the CP privileged command MONITOR to control monitoring, including the type, amount, and nature of data to be collected.
2	An application program such as Performance Toolkit for VM connects to the CP *MONITOR System Service to establish a data link with CP.
3	The monitor collects performance data during CP operation and stores it, in the form of monitor records, in a saved segment (MONDCSS). Note: MONDCSS is already defined for you during system installation.
4	An application program, such as Performance Toolkit for VM, can read data from the saved segment to display real-time performance characteristics. Performance Toolkit for VM can also process monitor data records written to disk or tape. While processing monitor data, Performance Toolkit for VM can generate historical monitor data that it can process at a later time.

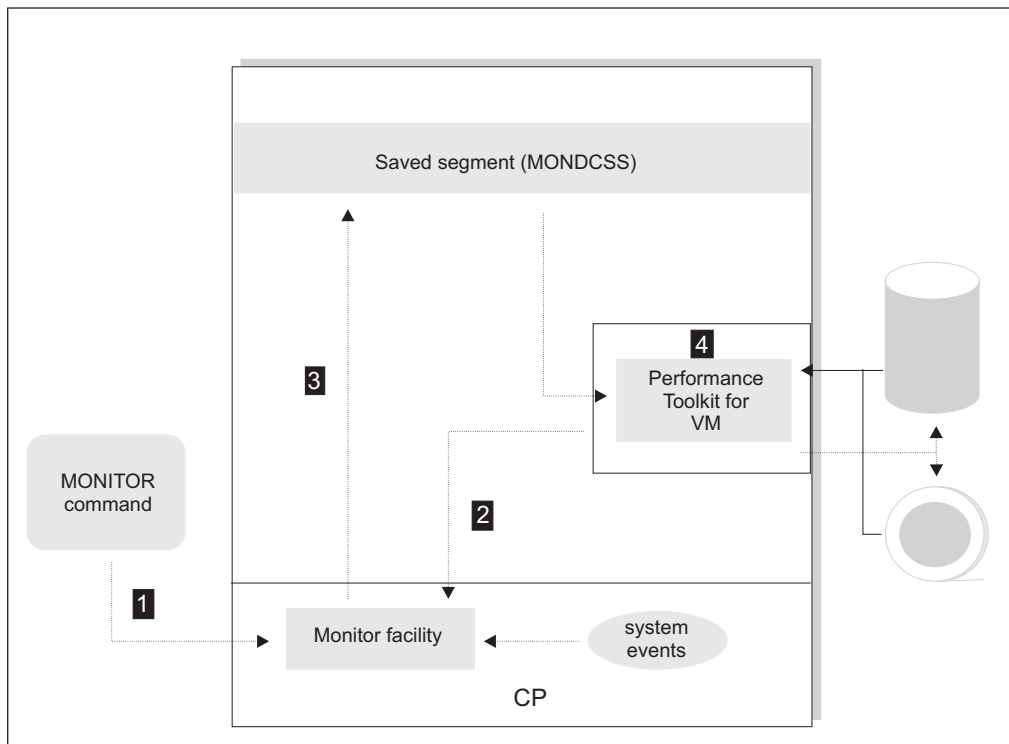


Figure 15. General monitoring process

Configuring Performance Toolkit for VM

Though Performance Toolkit for VM comes with z/VM, it is an optional feature in a disabled state. If you decide to use this feature, you must place an order for it so that you can enable it. This document assumes you have installed Performance Toolkit for VM according to *Program Directory for Performance Toolkit for VM*. Additional setup is required.

Through this task, you configure Performance Toolkit for VM on each system in the SSI cluster. For detailed information, see “Remote Performance Monitoring Facility” in *z/VM: Performance Toolkit Guide*.

To set up Performance Toolkit for VM, you must perform these subtasks:

Table 17. Tasks for setting up Performance Toolkit for VM

Subtask	Associated instructions (see . . .)
Setting up the performance analysis virtual machine (PERFSVM)	
<ul style="list-style-type: none"> Installing Performance Toolkit for VM 	<ul style="list-style-type: none"> <i>Program Directory for Performance Toolkit for VM</i> at IBM: z/VM Program Directories (www.ibm.com/vm/progdir/)
<ul style="list-style-type: none"> Configuring Performance Toolkit for VM 	<ul style="list-style-type: none"> “Steps for configuring Performance Toolkit for VM” “Steps for checking your Performance Toolkit for VM configuration” on page 159
(Optional) Setting up your Linux virtual servers to be monitored by Performance Toolkit for VM Note: To monitor your Linux virtual servers through Performance Toolkit for VM, you might need to install additional software on your Linux systems. See “Monitoring Linux virtual servers with Performance Toolkit for VM” on page 148.	“(Optional) Steps for setting up your Linux virtual servers to be monitored by Performance Toolkit for VM” on page 160
(Optional) Setting up the web interface for Performance Toolkit for VM	“(Optional) Steps for setting up the web interface for Performance Toolkit for VM” on page 161
(Optional) Testing your monitoring setup	“(Optional) Steps for testing your web interface setup” on page 162

Steps for configuring Performance Toolkit for VM

Before you begin: You need to install Performance Toolkit for VM by following the *Program Directory for Performance Toolkit for VM*. See IBM: z/VM Program Directories (www.ibm.com/vm/progdir/).

You need to log on to PERFSVM.

Perform these steps to configure Performance Toolkit for VM:

- Exit Performance Toolkit for VM by pressing PF3 twice.
- Remove the comments from the MONITOR commands in PERFSVM's PROFILE EXEC.
 - Edit the PROFILE EXEC. From the command line, type this command and press the Enter key:

xedit profile exec a
 - Remove the “/*” and “*/” characters that surround the MONITOR commands you need.

Example: The following PROFILE EXEC for PERFSVM enables monitor sampling for the typical samples and events that you need:

```
PROFILE EXEC      A1 V 130 Trunc=130 Size=16 Line=0 Col=1 Alt=0
```

```
00000 * * * Top of File * * *
00001 'CP MONITOR SAMPLE ENABLE PROCESSOR'
00002 'CP MONITOR SAMPLE ENABLE STORAGE'
00003 'CP MONITOR SAMPLE ENABLE USER ALL'
00004 'CP MONITOR SAMPLE ENABLE I/O ALL'
00005 'CP MONITOR SAMPLE ENABLE NETWORK'
00006 'CP MONITOR SAMPLE ENABLE APPLDATA ALL'
00007 'CP MONITOR SAMPLE ENABLE ISFC'
00008 'CP MONITOR SAMPLE ENABLE SSI'
00009
00010 'CP MONITOR EVENT ENABLE STORAGE'
00011 'CP MONITOR EVENT ENABLE I/O ALL'
00012 'CP MONITOR EVENT ENABLE NETWORK'
00013 'CP MONITOR EVENT ENABLE ISFC'
00014 'CP MONITOR EVENT ENABLE SSI'
00015
00016 'CP MONITOR SAMPLE INTERVAL 1 MIN'
00017 * * * End of File * * *
```

Note: Place all MONITOR commands before the PERFKIT command.

- c. Save the file. From the XEDIT command line, type this command and press the Enter key:

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

3. Copy the FCONX \$PROFILE to the 191 A-disk. Issue this command:

```
copy fconx $profile d = = a
Ready;
```

4. Edit FCONX \$PROFILE A.

- a. From the command line, type this command and press the Enter key:

```
xedit fconx $profile a
```

- b. Type the following XEDIT command and press the Enter key:

```
====> case mixed ignore
```

- c. Find the line with “MONCOLL VMCF”. From the XEDIT command line, type this command and press the Enter key:

```
====> /moncoll vmcf
```

- d. Replace the asterisk (*) in column 1 with an “F” for the “MONCOLL VMCF” statement:

```
FC MONCOLL VMCF ON
```

- e. Save the FCONX \$PROFILE. From the XEDIT command line, type this command and press the Enter key:

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


5. Create an FCONRMT SYSTEMS file that links your z/VM system and PERFSVM to special resource names. The resource names are used for remote access and are predefined, but you have to add them to the FCONRMT SYSTEMS file.

- a. Edit FCONRMT SYSTEMS A. From the command line, type this command and press the Enter key:

```
xedit fconrmt systems a
```

- b. Enter input mode. From the XEDIT command line, type this command and press the Enter key:

```
====> input
```

- c. Type the following lines and press the Enter key twice:

```
FCONRMT  SYSTEMS  A1  F 80  Trunc=80 Size=1 Line=1 Col=1 Alt=1
DMSXMD587I XEDIT:
```

```
* * * Top of File * * *
*Node-id Userid VM-Type Append Nickname
node_id1 PERFSVM z/VM6.2 N FCXC1R01
node_id2 PERFSVM z/VM6.2 N FCXC1R02
node_id3 PERFSVM z/VM6.2 N FCXC1R03
node_id4 PERFSVM z/VM6.2 N FCXC1R04
```

where *node_id1*, *node_id2*, and so forth are the names of your SSI members.

- d. Save the file. From the XEDIT command line, type this command and press the Enter key:

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

6. Set up a resource override for the default resource name.

- a. Edit UCOMDIR NAMES. Type the following command and press the Enter key:

```
xedit ucomdir names
```

- b. At the XEDIT command line, type the following command and press the Enter key:

```
====> input
```

- c. Type the following lines and then press the Enter key twice:

```
:nick.FCXRES00 :luname.*IDENT
               :tpn.FCXC1R01
               :security.SAME
:nick.FCXSYSTM :luname.*IDENT
               :tpn.FCXC1S01
               :security.SAME
```

- d. Save the file. From the XEDIT command line, type the following command and press the Enter key:

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

7. Create an FCONRMT AUTHORIZ file that makes PERFSVM a store and forward server. Additionally, configure PERFSVM to allow it to issue CP and CMS commands.

- a. Edit FCONRMT AUTHORIZ A. From the command line, type this command and press the Enter key:

```
xedit fconrmt authoriz a
```

- b. Enter input mode. From the XEDIT command line, type this command and press the Enter key:

```
====> input
```

- c. Type the following lines and then press the Enter key twice::

```
FCONRMT AUTHORIZ A1 F 80 Trunc=80 Size=18 Line=2 Col=1 Alt=2
DMSXMD573I Input mode:
```

```
* * * Top of File * * *
node_id1 PERFSVM S&FSERV DATA
node_id1 PERFSVM CMD
node_id2 PERFSVM CMD DATA
node_id3 PERFSVM CMD DATA
node_id4 PERFSVM CMD DATA
```

where *node_id1*, *node_id2*, and so forth are the names of your SSI members.

- d. Save the file. From the XEDIT command line, type this command and press the Enter key:

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

8. Check that AUTOLOG1's PROFILE EXEC automatically logs PERFSVM on. See the instructions in "Steps for automatically starting Linux virtual servers and other virtual machines" on page 107.

9. Perform these steps on each member of the SSI cluster, but code the FCONRMT AUTHORIZ and UCOMDIR NAMES files on each member differently.

- a. On the other SSI members, create an FCONRMT AUTHORIZ file that makes PERFSVM a store and forward server on that member. For example, on the second SSI member, FCONRMT AUTHORIZ would be:

```
FCONRMT AUTHORIZ A1 F 80 Trunc=80 Size=18 Line=2 Col=1 Alt=2
DMSXMD573I Input mode:
```

```
* * * Top of File * * *
node_id2 PERFSVM S&FSERV DATA
node_id2 PERFSVM CMD
node_id1 PERFSVM CMD DATA
node_id3 PERFSVM CMD DATA
node_id4 PERFSVM CMD DATA
```

where *node_id2* was the member name of the second member.

- b. Update the UCOMDIR NAMES file on each member to redefine the resource. For example, on the second SSI member, UCOMDIR NAMES would be:

```
:nick.FCXRES00 :luname.*IDENT
               :tpn.FCXC1R02
               :security.SAME
:nick.FCXSYSTM :luname.*IDENT
               :tpn.FCXC1S02
               :security.SAME
```

Resources for the third and fourth members are as follows:

Table 18. Resource names for UCOMDIR NAMES file

Member 3	Member 4
FCXC1R03 FCXC1S03	FCXC1R04 FCXC1S04

Stay logged on to PERFSVM and continue with the next instructions.

Steps for checking your Performance Toolkit for VM configuration

Before you begin: You need to be logged onto PERFSVM.

Perform these steps to check the configuration:

1. From the command line, run PERFSVM's PROFILE EXEC to start Performance Toolkit for VM. From the command line, type this command and press the Enter key:

```
profile
```

Result: You see the basic mode screen.

```
FCX001                Performance Toolkit for VM                Autoscroll 12
FCXBAS500I Performance Toolkit for VM FL620
FCXAPP530I Connected to *IDENT for resource FCXRES00
FCXAPP530I Connected to *IDENT for resource FCXSYSTM
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

```
Command ==>
F1=Help  F2=Redisplay  F3=Quit  F12=Return
```

2. From the basic mode screen, issue this command:

```
monitor
```

Result:

FCX124 Performance Screen Selection (FL620 PITS353) Perf. Monitor		
General System Data	I/O Data	History Data (by Time)
1. CPU load and trans.	11. Channel load	31. Graphics selection
2. Storage utilization	12. Control units	32. History data files*
3. SSI data menu*	13. I/O device load*	33. Benchmark displays*
4. Priv. operations	14. CP owned disks*	34. Correlation coeff.
5. System counters	15. Cache extend. func.*	35. System summary*
6. CP IUCV services	16. Reserved	36. Auxiliary storage
7. SP00L file display*	17. DASD seek distance*	37. CP communications*
8. LPAR data	18. I/O prior. queueing*	38. DASD load
9. Shared segments	19. I/O configuration	39. Minidisk cache*
A. Shared data spaces	1A. I/O config. changes	3A. Storage mgmt. data*
B. Virt. disks in stor.		3B. Proc. load & config*
C. Transact. statistics	User Data	3C. Logical part. load
D. Monitor data	21. User resource usage*	3D. Response time (all)*
E. Monitor settings	22. User paging load*	3E. RSK data menu*
F. System settings	23. User wait states*	3F. Scheduler queues
G. System configuration	24. User response time*	3G. Scheduler data
H. VM Resource Manager	25. Resources/transact.*	3H. SFS/BFS logs menu*
	26. User communication*	3I. System log
Select performance screen with cursor and hit ENTER		
Command ==>		
F1=Help F4=Top F5=Bot F7=Bkwd F8=Fwd F12=Return		

- Wait to allow Performance Toolkit for VM to collect data. Performance Toolkit for VM will not perform most screen calculations until at least two monitor intervals have passed. At least two monitor intervals are required to provide the delta for certain fields. When available, options are highlighted.
- Press the PF12 key to return to basic mode.

You are done when you know Performance Toolkit for VM is working. If you plan to follow the next instructions, stay logged onto PERFSVM.

(Optional) Steps for setting up your Linux virtual servers to be monitored by Performance Toolkit for VM
Before you begin: You need to be logged on to PERFSVM. You need to have the appropriate monitoring software installed on your Linux operating systems (see “Monitoring Linux virtual servers with Performance Toolkit for VM” on page 148).

Perform these steps to set up your Linux virtual servers to be monitored by Performance Toolkit for VM:

- Edit the FCONX LINUXUSR file. Issue this command:

```
xedit fconx linuxusr a
```

- Add the user IDs and IP addresses for the Linux virtual servers you want to monitor.
Example:

```
FCONX    LINUXUSR A1   F 80   Trunc=80 Size=5 Line=4 Col=1 Alt=0

* * * Top of File * * *
LINUX01   190.10.10.115:8803
LINUX02   190.10.10.116:8803
LINUX03   190.10.10.117:8803
LINUX04   190.10.10.118:8803
LINUX05   190.10.10.119:8803
```

- Save the file. From the XEDIT command line, issue:

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

Stay logged onto PERFSVM and continue with the next procedure.

(Optional) Steps for setting up the web interface for Performance Toolkit for VM

Before you begin: You need to be logged on to PERFSVM. Later, you need to log onto TCPMAINT.

Note: If you want to set up secure (SSL based) operation in order to safeguard passwords and data traveling across internet nodes outside your own area of administration, see “Internet Interface” in *z/VM: Performance Toolkit Guide*.

Perform these steps to set up the web interface:

1. Edit FCONX \$PROFILE A.

- a. From the command line, type this command and press the Enter key:

```
xedit fconx $profile a
```

- b. Find the line with “MONCOLL WEBSERV ON TCPIP”. From the XEDIT command line, type this command and press the Enter key:

```
====> /MONCOLL WEBSERV ON TCPIP
```

- c. Replace the asterisk (“*”) in column 1 with an “F” for the following statements:

```
FC MONCOLL WEBSERV ON TCPIP TCPIP 81  
FC MONCOLL LINUXUSR ON TCPIP TCPIP
```

- d. Save the FCONX \$PROFILE. From the XEDIT command line, type this command and press the Enter key:

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

2. Log onto TCPMAINT.

3. Edit the PROFILE TCPIP. Issue this command:

```
xedit profile tcpip
```

4. Find the PORT statement. For the supplied PROFILE TCPIP, the PORT statement follows the comment section beginning with “Reserve ports for specific server machines.” From the XEDIT command line, type this command and press the Enter key:

```
====> /Reserve ports for specific server machines.
```

5. Add port 81 to the PORT statement:

```

PROFILE  TCP/IP      A1  F 80  Trunc=80 Size=7 Line=5 Col=1 Alt=0

; Reserve ports for specific server machines.  Port values used are
; those defined in RFC 1060, "Assigned Numbers"
; -----
; Notes:
;
; - The TCP/IP stack restricts all "Well-Known" ports (ports 1-1023)
;   from general use on a default basis.
;
; - The FREELOWPORTS operand of the ASSORTEDPARMS statement can be
;   used to allow unrestricted use of well-known ports that have not
;   been otherwise reserved.  Alternatively, an asterisk (*) can be
;   specified in place of a conventional user ID for a PORT statement
;   entry to reserve a specific port number (or port range) for general
;   use.
; -----

PORT
20  TCP FTPSERVE  NOAUTOLOG ; FTP Server
21  TCP FTPSERVE           ; FTP Server
23  TCP INTCLIEN           ; TELNET Server
:
:
81  TCP PERFSVM           ; Performance Toolkit for VM SERVER

```

6. Save the file. From the XEDIT command line, type this command and press the Enter key:

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

7. Because TCP/IP is running, dynamically add the port association. From the command line, issue:

```

netstat obey port 81 tcp perfsvm
VM TCP/IP Netstat Level 620          TCP/IP Server Name: TCP/IP

09:24:06 * MSG FROM TCP/IP : OBEY PORT 81 TCP PERFSVM issued by TCPMAINT;
return code 0
RDR FILE 0032 SENT FROM TCP/IP      PRT WAS 0008 RECS 0002 CPY 001 A NOHOLD NOKEEP
OBEY command response is: OK
OBEY return code = 0

```

You are done when you receive return code 0 from NETSTAT OBEY.

(Optional) Steps for testing your web interface setup

Before you begin: You need be able to log off, then log back onto PERFSVM. You need to have a web browser and know the host IP address for z/VM. You can get the address from Performance Toolkit for VM's basic screen initialization panel.

Example:

```
FCXWWW587I Userid PERFSVM identified on 9.60.67.210:04157
```

Perform these steps to check your monitoring setup:

1. To re-initialize port 81:
 - a. Log off PERFSVM.
 - b. Log on PERFSVM.
2. Open your web browser to the z/VM IP address and port number for Performance Toolkit for VM.

3. On the Web Server Login screen, type your user ID and password.
4. Click the name of your z/VM host system.

Result: You see a screen like the following:

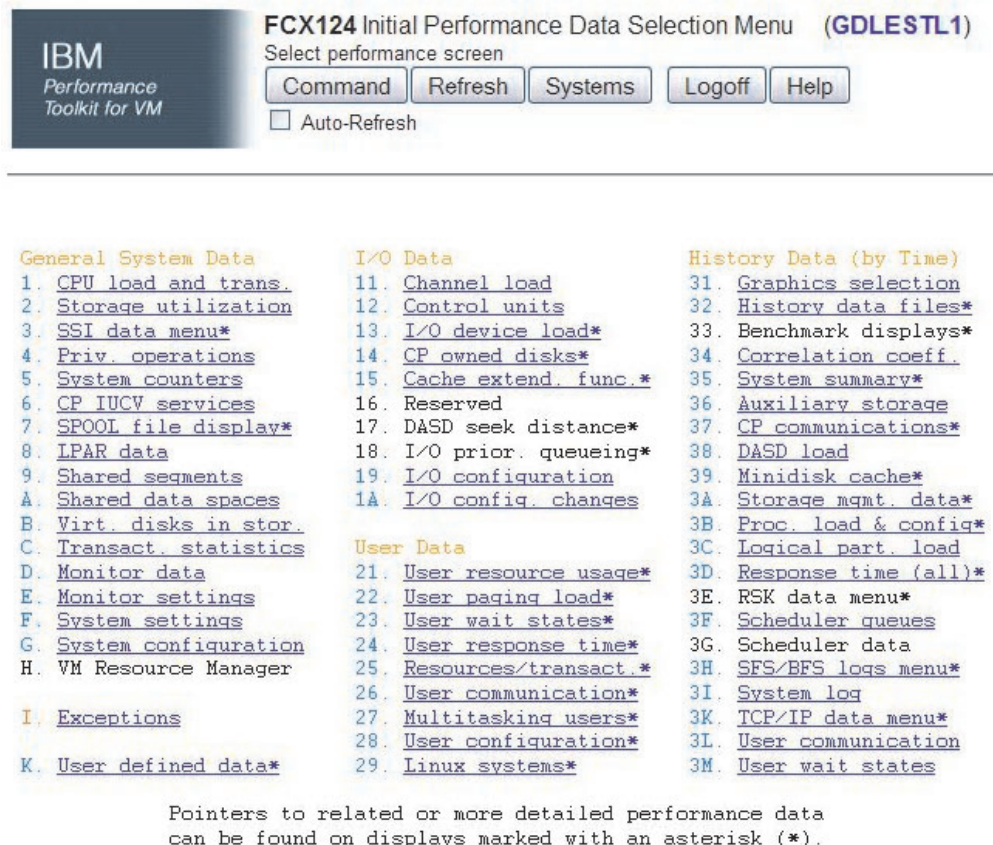


Figure 16. Initial Performance Data Selection Menu

5. Click “29. Linux Systems*” to see if the interface is displaying performance data from the Linux systems you configured.

Tips:

- A menu item is highlighted when data is available for that item.
- Use the navigation buttons and links in the web interface rather than your browser back and forward buttons.

You are done when you see the Linux performance data.

Using monitoring to analyze performance and capacity

This topic gives you a start on how to analyze performance and capacity problems. The topic does not cover performance and capacity in detail.

Related information:

- For more information on z/VM performance, see *z/VM: Performance*, SC24-6208.
- For detailed information about Performance Toolkit for VM screens, see *z/VM: Performance Toolkit Reference*, SC24-6210.

Steps for analyzing performance and capacity

Before you begin: You need to set up Performance Toolkit for VM and the web interface for Performance Toolkit for VM. See “Configuring Performance Toolkit for VM” on page 154.

Perform these steps to analyze performance and capacity:

1. Open your web browser to the z/VM host and port for Performance Toolkit for VM.
2. Base your next action on this table.

Tip: For an explanation of the columns in a panel, click the column heading.

Table 19. Decision table for analyzing performance and capacity

If you want to check for possible ...	Then check these screens ...	Notes
Storage constraints	For the whole system:	
	2. Storage utilization	Presents overall storage information for the z/VM member.
	For Linux guests:	
	21. User resource usage	
	22. User paging load	The screen shows detailed information about the virtual machine's paging load.
	29. Linux systems, then “LXMEM <i>userid</i> ” where <i>userid</i> is the virtual machine user ID	If you look at the details of a particular virtual machine you see the storage details for that virtual machine.
		Check “Swap-in rate” and “Swap-out rate.”
I/O constraints	11. Channel load	Busy channel paths on this screen can be an indication of I/O performance bottlenecks.
	13. I/O device load	This screen indicates I/O device throughput.
CPU constraints	For the whole system:	
	1. CPU load and trans.	Check the line: PROC %CPU %CP %EMU %WT %SYS %SP %SIC %LOGLD <ul style="list-style-type: none"> • “%LOGLD” is the best way to identify CPU bottlenecks. • “%CPU” through “%SP” are calculated on elapsed time and are a meaningful source for capacity planning.
	For Linux guests:	
	1. CPU load and trans.	Check “User Extremes:” (in the lower right of the panel) to identify heavy users or malfunctions of Linux guests (for example, looping).
	29. Linux systems	Check the Linux CPU utilization by entering “Linux <i>userid</i> ” (<i>userid</i> is the virtual machine user ID).
	21. User resource usage	Linux tools such as <i>top</i> report CPU consumption as if Linux owned the processors and do not consider that Linux is running under z/VM, so consumption figures differ with Performance Toolkit for VM.

Table 19. Decision table for analyzing performance and capacity (continued)

If you want to check for possible ...	Then check these screens ...	Notes
Unwanted resource consumption	1. CPU load and trans.	<ul style="list-style-type: none"> On Linux, use <code>ps -ef</code> to show the running tasks. Remove those tasks you do not need. Install the on-demand timer patch, which disables the Linux wake-up function and allows z/VM to determine when a Linux guest is truly idle.

You are done.

Steps for using CP commands to improve performance

You can use certain CP commands to aid the performance of Linux guest operating systems. Remember that improving the performance of one machine can impair the performance of others.

Note

Do these activities only from a user ID for which QUICKDSP is set on. You can determine whether you have QUICKDSP on by issuing `%CP QUERY QUICKDSP userid`; look for the QUICKDSP setting in the response. To set QUICKDSP on, issue the `%CP SET QUICKDSP userid ON` command.

Before you begin: You need be able to log on as MAINT.

Perform these steps to use CP commands to enhance performance:

1. Log on as MAINT.
2. Base your action on the choices in the following table.

Note: You can control other members in the SSI cluster by using the AT command.

Example:

```
at member2 cmd set share linux0 absolute 20%
```

Table 20. Decision table for using CP commands to improve performance

If you want to ...	Then use this command ...
Control the fraction of system resources a virtual machine receives.	<p>SET SHARE</p> <p>Example: By using these commands, you are establishing the fraction of system resources to which each virtual machine is entitled.</p> <pre>cp set share linux0 absolute 20% cp set share linux1 relative 100 cp set share linux2 relative 300 cp set share linux3 relative 100</pre> <p>Notes:</p> <ol style="list-style-type: none"> 1. A virtual machine receives its proportion of any scarce resource (CPUs, real storage, or paging I/O capability) according to its SHARE setting. 2. The SET SHARE command can be used to set target minimum and maximum values (notice that LINUX0 has a target minimum of 20%—that means other virtual machines contend for the remaining 80% of resources). 3. You can assign shares of system resources with the user directory SHARE statement so that when a user logs on, that virtual machine automatically has its share established.
Designate virtual machines that do not wait in the eligible list when they have work to do	<p>SET QUICKDSP <i>userid</i></p> <p>Example:</p> <pre>cp set quickdsp linux0 on USER LINUX0 : QUICKDSP = ON</pre> <p>Note: With this setting, virtual machines are assigned an eligible list class of E0 and are added to the dispatch list immediately. QUICKDSP overrides the scheduler's usual resource assessment and fitting algorithms for the target virtual machine and the virtual machine is run without regard to its resource needs. When you use QUICKDSP, you take responsibility for resource allocation. Set QUICKDSP on only for your most critical Linux servers to give them priority over other servers.</p>
Entitle a guest to a certain number of pages that should always be resident	<p>SET RESERVED</p> <p>Example:</p> <pre>cp set reserved linux01 50</pre> <p>means you are entitling the LINUX01 virtual machine to have 50 pages always resident in real storage.</p> <p>Note: Normal system stealing procedures may take any resident private page in any private address space owned by the user's virtual machine, down to the number of pages that are reserved (<i>nn</i>). Unlike the CP LOCK command, SET RESERVED does not guarantee that a specific resident page will remain resident. While the number of resident pages in a virtual machine is less than or equal to <i>nn</i>, normal system steal procedures will not affect it.</p>

You know you are done when system performance improves.

Related information:

For advanced information about performance and the z/VM scheduler, see

- *z/VM: Performance*, SC24-6208
- “VM Performance Resources” (www.vm.ibm.com/perf/)
- “The VM/ESA Scheduler Made Simple” (www.vm.ibm.com/devpages/bitner/presentations/vmsched.html)

Chapter 12. Servicing z/VM

This topic introduces IBM service concepts. After this introduction, turn to “Installing preventive or corrective service” in *z/VM: Service Guide*, GC24-6247 to install z/VM service.

z/VM service concepts

A component of z/VM called VMSES/E helps you install z/VM and other VMSES/E-enabled products and apply service changes that correct or circumvent reported problems. VMSES/E handles both source code and binaries.

Service is the process of changing a particular release of a software product. There are a number of reasons for servicing a product, such as:

- Correcting a problem. Problems that you report to IBM are first entered into *problem management records (PMRs)*. If IBM determines that a PMR requires a fix to a product, IBM creates a record called an *authorized program analysis report (APAR)*. APARs provide a formal method of tracking problems for a specific version of a product. An APAR can also affect several releases of a product. IBM fixes these problems for a particular release through *program temporary fixes (PTFs)*. A PTF contains the code changes for a solution to a problem (APAR) on a particular release. Each release of a product has a unique set of PTFs, because the fixes may be different on each release. When IBM ships a new release of a product like z/VM, PTFs from the previous release are merged into the new release.
- Adding function. New functions can be delivered in a new release or version of a product, or as service. When new function is delivered as service, it is called a *small programming enhancement (SPE)*. SPEs are delivered and tracked the same way as problems. An APAR is assigned to the SPE, and it is delivered as a program temporary fix (PTF).

IBM delivers service through:

- *Recommended service upgrade (RSU)* tapes, electronic envelopes, or CD-ROMs, which contain a collection of PTFs that IBM thinks are important enough that everyone should apply them. An RSU defines a *service level* for your product and is designed to prevent problems from occurring. (VMSES/E creates a software inventory and tracks the currency of the system through service levels.) This type of service is called preventive service.
- *Corrective service (COR)* tapes, electronic envelopes, or CD-ROMs, which contain PTFs that you request for a specific problem. Sometimes you cannot wait for an RSU to correct a problem, so you can order a PTF through a COR. This type of service is called corrective service because it is designed to correct a specific problem you encountered with the product.
- *Expanded Service Options (ESOs)*, which are defined collections of PTFs delivered in VMSES/E corrective service format. ESO allows you to choose the starting and ending service levels.

A product can have either an RSU or an ESO, but not both.

Other reasons for changing a product are:

- Circumventing a problem. For expediency, IBM may provide a circumvention for a problem until a PTF can be developed. A circumvention is meant to be a temporary solution to the problem, and it may be in the form of a procedural or software change. The method of delivery depends on the form of the solution,

and it is determined by you and the IBM support center. Once an APAR number is established for the problem, you can use that number to track the fix and see when a PTF is available.

- Applying local service or modifications. *Local service* and *local modifications* are defined as any service or software change that is applied to your z/VM system that was not supplied by IBM through corrective service (COR) or a recommended service upgrade (RSU).

When it is absolutely necessary to apply service from IBM before it is available on a COR tape, or when you need a local modification to tailor your system environment, you must apply the service locally. Local service includes updates supplied to you by other vendors or Licensed Products. Note that as subsequent PTFs are installed, both circumventions and local modifications may require rework.

For more information on adding local modifications, see “Applying local service and local modifications” in *z/VM: Service Guide*, GC24-6247.

In general, then:

- Someone reports a problem to IBM, which is entered as a problem management record (PMR).
- If it determines that a fix is needed, IBM creates an authorized program analysis report (APAR) for the code defect (bug).
- The solution for an APAR is a program temporary fix (PTF), which is like a code patch. The PTF might provide source updates in addition to new binaries.
- The PTF might be delivered as preventive service on a recommended service upgrade (RSU). RSUs are periodic and intended for everyone to apply as a way of preventing problems.
- For problems you encounter that require immediate fixes, the PTF can be delivered by corrective service (COR).
- Though PTFs usually contain fixes, sometimes IBM delivers new function through them, called small programming enhancements (SPEs).

Related Information:

- If you need to install service, follow the instructions in “Installing preventive or corrective service” in *z/VM: Service Guide*, GC24-6247.
- For details on the service process, see “VM Service Concepts” in *z/VM: VMSES/E Introduction and Reference*, GC24-6243.

Chapter 13. Preparing for live guest relocation

There are several reasons why you might need to relocate a virtual server while keeping the server available. You might need to relocate a virtual server for workload rebalancing, or to do software maintenance, or hardware maintenance. Before you relocate a guest, there are architectural, disk, memory, and networking requirements and guidelines you must understand.

What to know before starting relocations

CP does eligibility checking before a virtual server can relocate. You can check for relocation eligibility through the VMRELOCATE TEST command.

There are many requirements, guidelines, and restrictions for successful relocations. For a complete list, see “Conditions That Will Prevent a Relocation” in *z/VM: CP Planning and Administration*.

There are many other important factors to take into consideration before you start a relocation. This topic outlines some factors to consider for your configuration and provides some examples of how they might affect your relocations, as well as what you can do to avoid undesirable results.

Factors that can affect relocation are the following:

- Virtual machine size. The size of the virtual server's memory can affect total relocation time as well as quiesce time. During memory movement, CP attempts to relocate all the virtual server's memory in a series of passes, sending on each pass only the memory changed since the last pass. This process continues until internal algorithms determine no more progress can be made, at which point CP quiesces the virtual server. During quiesce time, CP must relocate the final guest state, I/O information, and changed pages. Thus, independent of other factors, the more memory a virtual server has, the longer the relocation and quiesce time.
- Virtual machine page change rate. The rate at which a virtual server changes its pages has a direct effect on total relocation time and, possibly, quiesce time. A guest changing pages rapidly has more pages to relocate in each memory pass and so the memory move stage lasts longer. A guest changing pages rapidly might also cause there to be more pages to move during quiesce time. Thus, independent of other factors, the higher a virtual server's page change rate, the greater its relocation time.
- ISFC set-up. All SSI clusters must have direct logical links between all systems. All SSI clusters use ISFC for intra-cluster communication and live guest relocation. ISFC uses CTC devices. For maximum throughput, when you are setting up your network, follow the guidelines in “Guidelines for planning your network in an SSI cluster” on page 40. Faster CTC speeds increase throughput and, independent of other factors, result in shorter relocation and quiesce times.
- Relocation options.
 - Synchronous (SYNC) and asynchronous (ASync) options. Synchronous relocations mean that only one relocation can be issued at a time by a particular user. Quiesce time is shortest when relocations are done synchronously, because the relocations are not competing with each other for system resources. Using the default (SYNC) option on the VMRELOCATE command can help ensure relocations are done serially, especially if you are issuing the VMRELOCATE command in an EXEC.

Using the ASYNC option on the VMRELOCATE command and issuing multiple relocations at once might cause undesirably long quiesce times.

- The IMMEDIATE option on the VMRELOCATE command causes quiesce time to occur after only one pass through memory. This option usually results in shorter overall relocation times, but longer quiesce times because the first pass through memory usually takes the longest, and the virtual server might have changed many pages, which then need to be moved during quiesce time. Longer quiesce times can cause undesirable results on applications that are running in a virtual server.
- Maximum total relocation time (MAXTOTAL) and maximum quiesce time (MAXQUIESCE) options. When these options are specified on the VMRELOCATE command, the relocation automatically cancels if either time limit is exceeded. The virtual server continues to run on the source system. The default value for MAXTOTAL is no limit, while the default for MAXQUIESCE is 10 seconds. You should investigate the applications running on the virtual server to determine the maximum amount of time the server can be unresponsive before the applications time out. Then, use this time to set the MAXQUIESCE time. The default 10 seconds might be too long for some applications, so setting your own MAXQUIESCE time is encouraged.
- Time for I/O relocation. While the virtual server is quiesced, I/O information is moved to the destination system. Before I/O information can be moved, all active I/O must complete. If any I/O errors occur, it might take longer for the active I/O to complete, which might cause undesirably long quiesce times.
- Other relocations occurring on the source and destination systems. Other relocations running to/from the source and destination systems create a load on those systems and on the links between those systems. This load might cause longer relocation or quiesce times. It is best to run all relocations within an SSI from one user, using the default SYNC option and the AT command to direct remote relocations.
- Other non-relocation activity on the source and destination systems. Constraints on the source or target system might make relocation or quiesce time longer and the addition of more virtual servers might have undesirable effects on the destination system.
- The SET RESERVED settings for a guest on the source member are not carried over to the destination member. If the virtual server had a SET RESERVED setting before relocating, reestablish the setting on the destination after the relocation completes, taking account of the available resources and workload on the destination system.
- SET SHARE settings are carried over to the destination system; however, the SET SHARE settings used on the source system might not be appropriate for the destination system. After relocation, reevaluate a virtual server's SET SHARE settings.

Related information: For more information about requirements for a successful guest relocation, see “Preparing for Live Guest Relocation in a z/VM SSI Cluster” in *z/VM: CP Planning and Administration*, SC24-6178.

Overview of relocation domains

A *relocation domain* defines a set of members of an SSI cluster among which virtual machines can relocate freely. A relocation domain can be used to define the subset of members of an SSI cluster to which a particular guest can be relocated. Relocation domains can be defined for business or technical reasons. For example, a domain can be defined that has all of the architectural facilities necessary for a

particular application, or a domain can be defined to allow access only to systems with a particular software tool. Whatever the reason for the definition of a domain, CP allows relocation among the members of the domain without any change to architectural characteristics or CP functionality as seen by the guest.

Relocation domains can be defined with the `RELOCATION_DOMAIN` system configuration statement or dynamically using the `DEFINE RELODOMAIN` command.

A virtual machine can be dynamically reassigned to a different relocation domain (or relocation can be enabled or disabled) with the `SET VMRELOCATE` command, but only if the new domain includes all of the architectural features in the current virtual architecture level (unless the `FORCE ARCHITECTURE` option is used).

Note: A virtual machine defined by a multiconfiguration virtual machine definition cannot be relocated and therefore a relocation domain cannot be set for it.

To determine the current relocation domain setting for a virtual machine logged on to the local member of the SSI cluster, use the `QUERY VMRELOCATE` command. (To determine the relocation domain setting for a virtual machine logged on to a remote member of the cluster, you can use the `AT` command to issue the `QUERY VMRELOCATE` command.) To find out what members of the SSI cluster are included in a particular relocation domain, use the `QUERY RELODOMAIN` command.

Related information: For a comprehensive introduction to relocation domains, see “Using Relocation Domains” in *z/VM: CP Planning and Administration*, SC24-6178.

Steps for defining relocation domains

Before you begin: Gather information about architectural levels, service requirements, and methods you use to apply hardware service.

Note: In a two member SSI cluster, all possible domains are by default predefined. This task assumes you have at least three cluster members and that you want to define custom relocation domains.

Perform the following steps:

1. Consider the architectural levels you have in your SSI cluster. If you have different architectural levels, you might form relocation domains accordingly. For example, if you have two members in your SSI cluster with and one system without cryptographic facilities, you might put the two systems with the cryptographic facilities together in a domain. All virtual servers in this domain would then be able to relocate only between members with cryptographic facilities.
2. Consider the service requirements you have and where you want to move virtual servers when you add service.
 - Look at the cluster member to which you wanted each virtual server to be able to relocate (see Table 4 on page 46).
 - When applying service, if you want to be able to move all the virtual servers from one cluster member to another, ensure the virtual servers can relocate without loss of facilities.
3. Consider how you apply hardware service. If you have two cluster members on one machine and you need to apply hardware service, you might want to move

all virtual servers from both of those members. Carefully consider all the possible permutations of service on your systems and plan your relocation domains accordingly.

4. After you plan your relocation domains, use the `RELOCATION_DOMAIN` statement to define them in your `SYSTEM CONFIG` file. For example:

```
RELOCATION_DOMAIN NORTH MEMBERS MEMBER1 MEMBER3
```

Notes:

- a. Do not use system identifiers to qualify `RELOCATION_DOMAIN` statements, because the domains should be the same throughout the SSI cluster.
 - b. It is a best practice to update the `SYSTEM CONFIG` file, so that if one or all members in the SSI cluster shut down and restart, the same relocation domains are defined.
5. Use the `DEFINE RELODOMAIN` command to define relocation domains dynamically. This command must be issued only on one system in the SSI cluster. For example:

```
define reلودomain north members member1 member3
Ready;
```

Note: The `DEFINE RELODOMAIN` command is automatically propagated to the other members of the SSI cluster.

6. Check your existing relocation domains with the `QUERY RELODOMAIN` command. For example:

```
query reلودomain all
DOMAIN NORTH    MEMBERS: MEMBER1  MEMBER3
DOMAIN MEMBER1  MEMBERS: MEMBER1
DOMAIN MEMBER2  MEMBERS: MEMBER2
DOMAIN MEMBER3  MEMBERS: MEMBER3
DOMAIN MEMBER4  MEMBERS: MEMBER4
DOMAIN SSI      MEMBERS: MEMBER1  MEMBER2  MEMBER3  MEMBER4
Ready; T=0.01/0.01 14:32:36
```

You are done when all your relocation domains are defined in the `SYSTEM CONFIG` and you have defined the domains dynamically.

Steps for placing Linux virtual servers into relocation domains

Before you begin: You should define your relocation domains. See “Steps for defining relocation domains” on page 171.

Note: In SSI clusters with only two members, you can relocate virtual servers only within the default SSI domain. The SSI domain operates at the lowest architectural level of any domain.

Perform the following steps:

1. From Table 4 on page 46, determine the members to which you want your virtual server to be able to relocate. Consider the architectural facilities required by that virtual server and ensure that all members you have chosen have access to those facilities.

2. Cross-reference your list of domains with your list of virtual servers and where you want them to relocate. Determine the relocation domain in which each virtual server should be. Put each virtual server in the smallest domain with the required architectural level.

For example, if a virtual server needs to relocate only between MEMBER1 and MEMBER2, put the virtual server in a domain containing only MEMBER1 and MEMBER2, not the SSI domain, which might have a lower architecture level than desired.

3. After you determine which domains virtual servers should be in, assign them to their domains in their directory entries. For example, type the following command and press the Enter key:

```
dirmaint for linux01 vmrelocate on domain north
DVHXMT1191I Your VMRELOCATE request has been sent for processing to
DVHXMT1191I DIRMAINT at MEMBER1 via DIRMSAT2.
Ready; T=0.01/0.02 14:37:04
DVHREQ2288I Your VMRELOCATE request for LINUX01 at
DVHREQ2288I * has been accepted.
DVHBIU3450I The source for directory entry LINUX01
DVHBIU3450I has been updated.
DVHBIU3424I The next ONLINE will take place
DVHBIU3424I immediately.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DSATCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHRLA3891I Your DMVCTL request has been relayed
DVHRLA3891I for processing.
DVHBIU3428I Changes made to directory entry
DVHBIU3428I LINUX01 have been placed online.
DVHREQ2289I Your VMRELOCATE request for LINUX01 at
DVHREQ2289I * has completed; with RC = 0.
```

4. If the virtual server is currently logged on, you can dynamically set the domain by command. For example:

```
set vmrelocate linux01 on domain north
Running on member MEMBER1
Relocation enabled in Domain NORTH
Ready; T=0.01/0.01 14:38:46
```

Notes:

- a. The previous step made the relocation domain setting permanent, which is what you should do so that the relocation domain setting persists after logging off.
 - b. A virtual server can be set to a domain operating at a higher level than its current domain at any time. A virtual server can not be set to a domain operating at a lower level than its current domain because doing so could cause a loss of facilities and potentially hurt virtual server operation.
5. Test the domain assigned to the virtual server with the QUERY VMRELOCATE *userid* command. For example:

```
query vmrelocate linux01
Running on member MEMBER1
Relocation enabled in Domain NORTH
Ready; T=0.01/0.01 14:38:46
```

6. Repeat these steps for all virtual servers that you want to be relocatable.

You are done when all virtual servers have been assigned to a relocation domain.

Steps for performing a relocation for a Linux virtual server

Before you begin: You should have your relocation domains defined and all relocatable virtual servers assigned to their proper domains.

You should have EQIDs assigned to all necessary devices (for example, OSAs, FCPs, and HiperSockets) connected to your relocatable virtual servers. See “Step for managing the SSI cluster” on page 133.

If your virtual server uses FCP devices, it should have multipathing support enabled through the “queue_if_no_path” option.

Perform the following steps:

1. Choose one class A user ID from which to issue all your relocations. Issuing the VMRELOCATE command from one user ID helps keep your relocations organized and helps you avoid issuing multiple relocations at once.
2. For each virtual server you wish to relocate, determine if there is a maximum amount of time the server can be quiesced. Look at the applications running on the virtual server to determine if there are any timeout values for those applications. For instance, a web server might need to reply to a request to serve a page within five seconds.
3. Issue the VMRELOCATE TEST command. The command tells you any reasons why the virtual server is not eligible to relocate. If there are any issues, address them and then retry the VMRELOCATE TEST command.

```
vmrelocate test linux01 member1
User LINUX01 is eligible for relocation to MEMBER1
Ready; T=0.01/0.01 12:00:09
```

4. Issue the VMRELOCATE MOVE command with the MAXTOTAL and MAXQUIESCE times that you determined.

By default the command is issued synchronously, meaning you can not issue any other commands while the VMRELOCATE MOVE is ongoing. This restriction is designed to help you serialize relocations, which is the best practice. However, you might use the ASYNC option on the command so you can issue other commands during the relocation, such as a VMRELOCATE STATUS to see how the relocation is proceeding.

```
vmrelocate move linux01 member1 maxtotal 10 maxquiesce 5
Relocation of LINUX01 from MEMBER2 to MEMBER1 started
User LINUX01 has been relocated from MEMBER2 to MEMBER1
```

You know you are done when your virtual server has been relocated.

Steps for troubleshooting guest relocations

You can track the process of guest relocation with the VMRELOCATE STATUS command. The process steps are:

- Connecting. The source system is connecting to the destination system.
- Eligibility checks. Relocation eligibility checking is in progress.
- Creating guest. Creating a skeleton guest on destination system.
- Moving memory. The virtual machine's memory is being transferred to the destination system.
- Quiesce time:
 - Stopping guest. The virtual machine is being stopped on the source system.
 - Moving guest. The virtual machine state is being moved to the destination system.
 - Final mem copy.
 - Final I/O check.
- Resume Guest. The virtual machine is being started on the destination system.
- Cleaning Up. The relocation is finished and cleanup work is being done.

To troubleshoot guest relocations, follow Table 21.

Table 21. Troubleshooting guest relocations

If you see this symptom...	Then do this...
A relocation is taking a really long time.	<ul style="list-style-type: none">• Issue VMRELOCATE STATUS <i>user</i> DETAILS periodically. This action shows the amount of memory being moved. If the number of pages moved and number of passes are changing, relocation is still working. Keep in mind that memory move is, especially for large guests, the longest stage in relocation. The guest is still running on the source, so there are no ill effects to the guest.• Consider adding more central storage (memory) to the source or target system or adding more FICON CTCs to help the relocation go faster.
Quiesce time is taking too long.	<ul style="list-style-type: none">• Issue a VMRELOCATE CANCEL <i>user</i> so the guest continues running on the source system.• If the destination system has a high paging rate and you have standby memory available, try adding more memory (central storage). If you do not have standby memory, you must shut down z/VM and redefine memory for the LPAR in which z/VM runs. Consult your hardware information about adding more memory to the LPAR.• If the destination system is running high CPU utilization rates, try adding more processors.• If your configuration is not at the specified guidelines for best performance, try adding more FICON CTCs, upgrading your existing CTCs to faster ones, or both. First run the cables to get the additional CTCs connected. Then activate the ISFC links on the target and source members by using the ACTIVATE ISLINK command.• When you re-try the relocation, be sure to specify MAXQUIESCE to ensure your quiesce time doesn't go too long again.

Table 21. Troubleshooting guest relocations (continued)

If you see this symptom...	Then do this...
A relocation was started with no MAXQUIESCE or MAXTOTAL value, but one is needed.	Use the VMRELOCATE MODIFY command to change the MAXQ or MAXT values.

More information: For more information on live guest relocation, see:

- *z/VM: CP Planning and Administration*
- *z/VM: CP Commands and Utilities Reference*
- *z/VM: Performance*

Appendix. Example of using FTP to install Linux from the hardware management console

If you do not have an external FTP server or do not want to create an external connection due to security concerns, you can install Linux by accessing a DVD or removable medium in the hardware management console (HMC) through FTP. You can link the hardware management console (HMC) DVD drive to your z/VM logical partition and ftp to the HMC DVD drive (using loopback FTP).

Notes:

1. This support is intended for customers who have no alternative, such as a LAN-based server, for serving the DVD contents for Linux installations. The elapsed time for installation using the HMC DVD drive can be an order of magnitude, or more, longer than the elapsed time for LAN-based alternatives.
2. If you need to avoid external connections, the z/VM FTP server must be connected through a guest LAN or VSWITCH to the virtual machine installing Linux. For more information, see *z/VM: TCP/IP Planning and Customization*.
3. To retrieve the initial boot files with FTP, you use the loopback address, but for the installation, when Linux asks for an FTP address to retrieve files, you use the home address of the z/VM TCP/IP server.
4. Because the directory containing the Linux boot files varies based on the Linux distribution and distribution level, the directory information used in the FTP GET commands in this topic is for example use only.

Linking the HMC removable media to your z/VM logical partition

Perform these steps to link the removable media in the HMC to the logical partition in which z/VM is running:

1. Open **CPC Images** on the HMC:
 - a. From the Views area, double-click the **Task List**.
 - b. From the Task List Work Area, open **CPC Recovery**.
 - c. From the Views area, open **Groups**.
 - d. From the Groups Work Area, open **CPC Images**.
2. Link the z/VM image to the HMC removable media:
 - a. From the CPC Images Work Area, drag the z/VM image you want to link to and drop it on the **Access Removable Media** icon in the CPC Recovery Area. The Access Removable Media panel opens.
 - b. On the Access Removable Media panel, click the appropriate option button to select the removable media that you want the image to access. Depending on which drives are available in the HMC, possible choices might include a DVD-RAM drive, USB flash memory drive, or diskette drive.

Note: For supported USB flash memory drives, see *System z Hardware Management Console Operations Guide*.

- c. Click **OK** on the Access Removable Media panel.
- d. Click on **OK** in the Access Removable Media Task Confirmation panel. The Access Removable Media panel is displayed.

Important: Keep the Access Removable Media panel open until you finish the transfer (clicking **OK** on the panel closes access to the removable media). Only when you are done using the removable media should you click **OK** to end access.

FTPing to the HMC removable media

Before using FTP, you must grant authority in the FTP server to the user ID that accesses the HMC removable media. This user ID is the one you enter when the z/VM FTP client prompts USER (identify yourself to the host):. This example uses the user ID SIMONW.

You must also establish the HMC removable media as the default (initial) directory for the user ID accessing the removable media. The z/VM FTP server recognizes the fixed string `../HMC:/` (case insensitive) to specify access to the removable media in the HMC. When there are multiple HMCs, the z/VM FTP server contacts the HMC that is linked to the LPAR using the Access Removable Media function described in “Linking the HMC removable media to your z/VM logical partition” on page 177.

These steps show how to grant authority and establish the initial directory. For the user ID accessing the HMC removable media, you must queue an HMCAUTH YES string followed by a `../HMC:/` string to the program stack in the z/VM FTP server CHKIPADR exit.

Perform these steps to ftp to the HMC removable media through the z/VM FTP server:

1. For the user ID accessing the HMC removable media, grant authority to ftp to the HMC removable media through the z/VM FTP server. To grant authority, xedit the CHKIPADR SAMPEXEC file on the TCPMAINT 198 disk.
 - a. Locate and remove the unconditional exit line (Exit 0).

```
/*-----*
/* Unconditional Exit - Simulate result of nonexistent exit.      *
/*-----*
/* To enable the SAMPLE STATEMENTS provided within this file (or to *
/* enable the execution of your customized statements and logic),  *
/* DELETE the "Exit 0" statement that follows this comment block.  *
/*-----*
Exit 0    /* delete this line */
```
 - b. Add the following case for the user ID that you use to access the removable media. For example, add the case for SIMONW:

```
When (Userid = 'SIMONW') Then
Do
    Queue 'HMCAUTH YES'
    Queue '../HMC:/'
End
```
 - c. Save the file as CHKIPADR EXEC.
 - d. Restart the z/VM FTP server.
2. For your A-disk, access a disk large enough to hold the boot files (the 192 disk in the example) and then ftp to the loopback address as follows:

Note: Parts in bold indicate commands you would issue.

```
access 192 a
DMSACC724I 192 replaces A (191)
Ready;
```

```
ftp 127.0.0.1
VM TCP/IP FTP Level 62
```

```
Connecting to 127...1, port 21
220-FTPSERVE IBM VM Level 62 at VM.DOMAIN.COM
220 Connection will close if idle for more than 5 minutes.
```

USER (identify yourself to the host):

```
simonw  
>>>USER simonw  
331 Password required for simonw.  
Password:  
>>>PASS <<<<<<<<  
230 User simonw logged in; working directory = ../HMC:/  
Command:
```

3. Continue with the FTP transfer. Note the command `locsite fix 80`, which sets the VM file format to fixed length 80, the file format necessary for punching the binary files to the virtual machine reader.

```
bin  
>>>TYPE i  
200 Type set to I.  
locsite fix 80  
Command:  
get /boot/s390x/vmrdr.ikr  
>>>PORT 127,0,0,1,4,5  
200 PORT command successful.  
>>>RETR /boot/s390x/vmrdr.ikr  
150 Opening BINARY mode data connection for vmrdr.ikr (1511884 bytes).  
226 Transfer complete.  
1511920 bytes transferred in 1.972 seconds. Transfer rate 766.69 Kbytes/sec.  
Command:  
get /boot/s390x/initrd suse.initrd  
>>>PORT 127,0,0,1,4,7  
200 PORT command successful.  
>>>RETR /boot/s390x/initrd  
150 Opening BINARY mode data connection for initrd (9862505 bytes).  
8299680 bytes transferred.  
226 Transfer complete.  
9862560 bytes transferred in 12.366 seconds. Transfer rate 797.55 Kbytes/sec.  
Command:  
asc  
>>>TYPE a  
200 Type set to A.  
Command:  
get /boot/s390x/parmfile parm.file  
>>>PORT 127,0,0,1,4,8  
200 PORT command successful.  
>>>RETR /boot/s390x/parmfile  
150 Opening ASCII mode data connection for parmfile (38 bytes).  
226 Transfer complete.  
40 bytes transferred in 0.081 seconds. Transfer rate 0.49 Kbytes/sec.  
Command:  
quit
```

Notices

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsurama, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

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

Adobe and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java is a trademark of Oracle and/or its affiliates.

Glossary

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

The z/VM glossary is also available through the online z/VM HELP Facility. For example, to display the definition of the term “dedicated device”, issue the following HELP command:

```
help glossary dedicated device
```

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

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

```
help glossary newterm
```

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

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

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

Bibliography

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

Where to Get z/VM Information

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

- z/VM V6R2 Information Center (publib.boulder.ibm.com/infocenter/zvm/v6r2/)
- IBM: z/VM Internet Library (www.ibm.com/vm/library/)
- IBM Publications Center (www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)
- IBM Online Library: z/VM Collection, SK5T-7054

z/VM Base Library

Overview

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

Installation, Migration, and Service

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

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6167
- *z/VM: CMS Planning and Administration*, SC24-6171
- *z/VM: Connectivity*, SC24-6174
- *z/VM: CP Planning and Administration*, SC24-6178
- *z/VM: Getting Started with Linux on System z*, SC24-6194
- *z/VM: Group Control System*, SC24-6196
- *z/VM: I/O Configuration*, SC24-6198
- *z/VM: Running Guest Operating Systems*, SC24-6228

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

Customization and Tuning

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

Operation and Use

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

Application Programming

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

- *z/VM: Enterprise Systems Architecture/Extended Configuration Principles of Operation*, SC24-6192
- *z/VM: Language Environment User's Guide*, SC24-6199
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6202
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6203
- *z/VM: OpenExtensions Commands Reference*, SC24-6204
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6205
- *z/VM: OpenExtensions User's Guide*, SC24-6206
- *z/VM: Program Management Binder for CMS*, SC24-6211
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6220
- *z/VM: REXX/VM Reference*, SC24-6221
- *z/VM: REXX/VM User's Guide*, SC24-6222
- *z/VM: Systems Management Application Programming*, SC24-6234
- *z/VM: TCP/IP Programmer's Reference*, SC24-6239
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- *z/OS: IBM Tivoli Directory Server Plug-in Reference for z/OS*, SA76-0148
- *z/OS: Language Environment Concepts Guide*, SA22-7567
- *z/OS: Language Environment Debugging Guide*, GA22-7560
- *z/OS: Language Environment Programming Guide*, SA22-7561
- *z/OS: Language Environment Programming Reference*, SA22-7562
- *z/OS: Language Environment Run-Time Messages*, SA22-7566
- *z/OS: Language Environment Writing Interlanguage Communication Applications*, SA22-7563
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *z/OS MVS Program Management: User's Guide and Reference*, SA22-7643

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6161
- *z/VM: CP Messages and Codes*, GC24-6177
- *z/VM: Diagnosis Guide*, GC24-6187
- *z/VM: Dump Viewing Facility*, GC24-6191
- *z/VM: Other Components Messages and Codes*, GC24-6207
- *z/VM: TCP/IP Diagnosis Guide*, GC24-6235
- *z/VM: TCP/IP Messages and Codes*, GC24-6237
- *z/VM: VM Dump Tool*, GC24-6242
- *z/OS and z/VM: Hardware Configuration Definition Messages*, SC33-7986

z/VM Facilities and Features

Data Facility Storage Management Subsystem for VM

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

Directory Maintenance Facility for z/VM

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

Open Systems Adapter/Support Facility

- *zEnterprise System, System z10, System z9 and eServer zSeries: Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7935
- *System z9 and eServer zSeries 890 and 990: Open Systems Adapter-Express Integrated Console Controller User's Guide*, SA22-7990

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

Performance Toolkit for VM

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

RACF Security Server for z/VM

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

Remote Spooling Communications Subsystem Networking for z/VM

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

Prerequisite Products

Device Support Facilities

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

Environmental Record Editing and Printing Program

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

Additional Publications

For white papers, IBM Redbooks publications, and other useful information about Linux on the mainframe, visit:

IBM: z/VM resources for Linux on IBM System z (www.vm.ibm.com/linux/)

Another source of Linux information is on the IBM developerWorks website at:

IBM developerWorks: Linux: Linux on System z: Documentation: Development Stream (www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

Index

Special characters

:AUTOBLOCK. 76
:DEFAULTS. 76
:EXCLUDE. 76
:GROUPS. 76
:Regions. 75
:SSI_VOLUMES. 76

A

A-disk (191) 14
access mode 14
ADD prefix command 23
archive
 Linux disks 141, 143
 system disks 141, 142
ASCII console 119
AUTHFOR CONTROL file 75
authorized program analysis report (APAR) 167
Auto_Warm_IPL 60
AUTOLOG1 83, 107
autologon
 Linux 107
 PERFSVM 108
 TCP/IP 83

B

backup 141
benchmarking 27
boot
 See initial program load (IPL)
boot files, Linux 99

C

capacity 27
checking the system 87
Clear_TDisk 60
cloning 105
CMS
 See Conversational Monitor System
CMS commands 14, 15
CMS files 15
CNTRL-C 137
CONFIG DATADVH file 74
configuration, sample 3
console
 See *also* virtual console
 addresses 65
 automating 115
 hardware management console 119
 logical operator 110, 119
 secondary 116
 system operator 119
 virtual 7, 119
Control Program 2, 5

Conversational Monitor System 3, 14
corrective service (COR) 167
CP
 See Control Program
CP commands 8
CP READ 7
CP-owned volume 52
CPSYNTAX command 68
CPU
 real and virtual 5
 requirements, real 32
 requirements, virtual 33
cryptographic facility 36, 95, 124, 139

D

Delete key 23
DELETE prefix command 23
devices, managing 122
direct access storage device (DASD)
 archiving 143
 definition of 6
 requirements 33
 restoring 144
directory entry
 cloning 37
 example 12
 Linux master 93
Directory Maintenance Facility
 6VMDIR20 virtual machine 73
 AUTHFOR CONTROL file 75
 authorizing users 75
 CONFIG DATADVH file 74
 configuring 71, 73
 enabling 71
 entering commands 81
 EXTENT CONTROL file 75
 group name 77
 help 127
 managing users 128
 minidisks, where created 75
 overview 37
 password 72
 putting into production 79
 region ID 77
 task roadmap 38
 testing 80
 USER DIRECT file 74
DirMaint
 See Directory Maintenance Facility
Disconnect_timeout feature 60
discontiguous saved segment (DCSS) 142, 153
disks
 See direct access storage device (DASD)
dispatch list
 INDICATE LOAD command 150
 overview 149
 SET QUICKDSP command 166

documentation, Linux 48
dormant list 149
DVD/removable media, installing Linux from 177

E

edit mode 21
editing screen 19
editor, XEDIT 19
eligible list
 definition of 149
 INDICATE LOAD command 150
 INDICATE QUEUES command 151
 SET QUICKDSP command 166
Emergency_Message_Console 66
envelope 167
escape character 67, 91
expanded service option (ESO) 167
expanded storage
 allocating 31
 attaching 126
 detaching 140
EXTENT CONTROL file 75

F

features 60, 89
file mode 15, 20
file name 15, 20
FILE subcommand 24
file type 15, 20
FILELIST command 15
files, editing 23, 24
FORCE command 132

G

guest LAN 35
guest operating system
 definition of 2
 disks 33, 57
 example 3
 memory 32
 network 34
 virtual switch 35, 63, 94, 101

H

hcp utility 138
Help
 commands 18
 messages 120
HMC (hardware management console)
 description 119
 DVD/removable media installation 177
 installing Linux 177
HOLDING 7

I

INDICATE command 150
initial program load (IPL)
 definition of 25
 loading Linux automatically 107
 z/VM, restarting 87, 122
input mode, editing 22
INPUT prefix command 23
Insert key 23
integrated ASCII console 119
ISFC links 40, 56

L

LINDFLT DIRECT 93
LINUX PROTODIR 93
Linux virtual server
 autologon 107
 boot files 99
 cloning 105
 creating first 93
 definition of xii
 disks 33
 installing, overview 99
 monitoring 148, 160
 shutdown 109
 virtual console 115, 119
 virtual switch 35, 63, 94, 101
live guest relocation
 Linux 172
 overview 4
 preparing 169
 relocation domain 169, 170, 171
 troubleshooting 175
load
 See initial program load (IPL)
local modification 168
local service 168
LOCATE subcommand 23
logical NOT 137
logical operator 110
logical partition (LPAR) 3

M

memory requirements
 real machine 31
 virtual machine 32
minidisk
 access mode 14
 disk sharing 6
 guidelines 33
 Linux disks 95
 MDISK statement 12
 search order 14
 user volume list 57
MONDCSS 153
monitor, overview 153
MORE ... 7
multiconfiguration virtual machine 11

N

- networking
 - options 34
 - virtual switch 35, 63, 90, 94
- NOT ACCEPTED 7

O

- Offline_at_ipl 62, 90
- operating system, virtual machine 1
- operations
 - real system 120
 - virtual machine 137
- OPERATOR
 - automation 110
 - loading CMS 81
 - logical operator 110
 - programmable operator 110
- operator consoles
 - monitoring 120
 - overview 119
 - setting addresses 65

P

- paging volume 33, 49, 121
- parm disk 52
- Passwords_on_Cmds feature 60
- PDR
 - See persistent data record
- performance monitoring
 - overview 147
 - snapshot, taking 150
 - using Performance Toolkit for VM 164
- Performance Toolkit for VM
 - autologon 108
 - checking configuration 159
 - setting up 154
 - using 164
- persistent data record (PDR) 39
- preventive service 167
- printer 7
- problem management record (PMR) 167
- PROFILE EXEC
 - AUTOLOG1 83, 107
 - overview 16
 - PERFSVM 155
 - programmable operator 115
- program temporary fix (PTF) 167
- programmable operator facility 110, 114
- prototype directory entry 37, 93
- punch 7

Q

- QQUIT subcommand 24
- QUERY *rdev* command 123
- QUERY *userid* command 132
- QUERY NAMES command 132
- QUERY PATHS command 125

- QUERY SHARE command 152
- QUERY USERS command 130, 132
- QUICKDSP option 95, 165, 166
- QUIT subcommand 24

R

- reader 7
- recommended service upgrade (RSU) 167
- registry, user
 - See user directory
- restarting z/VM 87, 122
- restoring disks 144
- Retrieve feature 60
- routing table 110, 111
- run-time tasks 119
- RUNNING 7

S

- SAVE subcommand 24
- saved segment 142, 153
- scheduler, overview 148
- search order, CMS 14
- secondary console 116
- service
 - corrective 167
 - preventive 167
- SERVICE DIRM ENABLE command 71
- service level 167
- service virtual machine 10
- shared segment 6
- shutdown, Linux 107, 109
- ShutdownTime 60, 89
- SIGNAL SHUTDOWN command 133
- Signal ShutdownTime 60, 89
- similarities, Linux and z/VM 25
- single system image cluster
 - overview 4
 - planning
 - CP-owned volumes 52
 - cryptographic feature 43
 - DASD 39
 - LPAR, memory, processor 38
 - network 40
 - saved segments 40
 - steps 44
 - tapes 43
 - users 43
 - virtual server networks 43
- single-configuration virtual machine 11
- small programming enhancement (SPE) 167
- special characters 67, 91
- spool file 9
- spool file commands 10
- spool file system 9
- spooling device 9
- spooling volume
 - adding to system 49, 52
 - checking 87
 - warning 121

SSI

See single system image cluster

status notices 7

storage (memory)

overview 6

real machine 31

virtual machine 32

system automation 107

SYSTEM CONFIG file 49, 68

system identifier 88

system performance 165

T

tasks

(noun, gerund phrase)

steps 91, 174

activating ISFC links

steps 56

adding CMS to LINMSTR

steps 97

Auto_Warm_IPL, updating 60

Clear_TDisk, enabling 60

console, setting addresses 65

CP commands at the Linux system console,
using 138

CP commands to enhance performance, using 165

CP commands to monitor performance, using 150

CP-owned volume list, updating 52

critical data, archiving 141

data, backing up 141

defining relocation domains

steps 171

devices

checking 90, 122

controlling at startup 62

managing 122

DirMaint

AUTHFOR CONTROL file, creating 75

configuring 71, 73

EXTENT CONTROL file, updating 75

password, updating 72

production, putting into 79

testing 80

USER DIRECT file, copying 74

Disconnect_timeout, setting 60

disks, restoring 144

documentation and media, obtaining 48

features list

checking 89

updating 60

Linux

automating console 115

creating first 93

setting up 191 disk 99

setting up monitoring 160, 161, 162

shutdown 109

starting automatically 107

network connection, setting 83

OPERATOR, modifying directory entry 81

paging volume, formatting 49

tasks (*continued*)

performance monitoring

configuring 154

overview 147

performance snapshot, taking 150

PERFSVM, setting up 155

placing virtual servers into relocation domains

steps 172

planning SSI cluster

steps 44

primary parm disk

releasing 52

programmable operator, setting up 109

retrieve buffers

checking 90

updating 60

ShutdownTime

checking 89

updating 60

Signal ShutdownTime

checking 89

updating 60

special characters

changing default 67

checking 91

spooling space

checking 87

updating 52

spooling volume, formatting 49

SSI cluster, planning steps 44

SYSTEM CONFIG, checking syntax 68

system identifier

checking 88

TCP/IP

autologon 83

checking 92

task roadmap 35

user volume list

checking 89

updating 57

virtual switch

checking 90

configuring 35, 63

z/VM, restarting 87

TCP/IP

autologon 84

checking 92

network connection 83

networking options 34

production 83

task roadmap 35

temporary minidisk 6, 49

U

unit record device 9

user directory

DirMaint 37, 71

overview 11

USER DIRECT file 74

user management 37

- user ID, VM 2
- user volume list 57, 89
- users, managing 37, 126

V

- VARY CHPID command 126
- VARY ONLINE/OFFLINE command 124, 125
- VARY PATH command 125
- virtual console
 - automating 115
 - overview 7, 119
 - status 7
- virtual CPU
 - adding to virtual machine 139
 - detaching 140
 - MACHINE statement 12
 - overview 5
 - requirements 33
- virtual disk in storage 6, 32
- virtual machine 1
- virtual server
 - See Linux virtual server
- virtual switch
 - checking 90
 - configuring 35, 63
 - overview 35
 - task roadmap 35
- virtual switch controller, overview 35
- vmcp command 138
- VMREAD 7
- volume, DASD 6

W

- warm start 60
- Web interface, Performance Toolkit for VM 161

X

- XAUTOLOG command
 - logging on a user 132
 - TCP/IP 84
- XEDIT 19

Z

- z/VM
 - restarting 87, 122



Product Number: 5741-A07

Printed in USA

SC24-6194-01

