Redbooks
ibm.com/redbooks

Virtualization Cookbook for IBM Z Volume 1

IBM z/VM 7.2

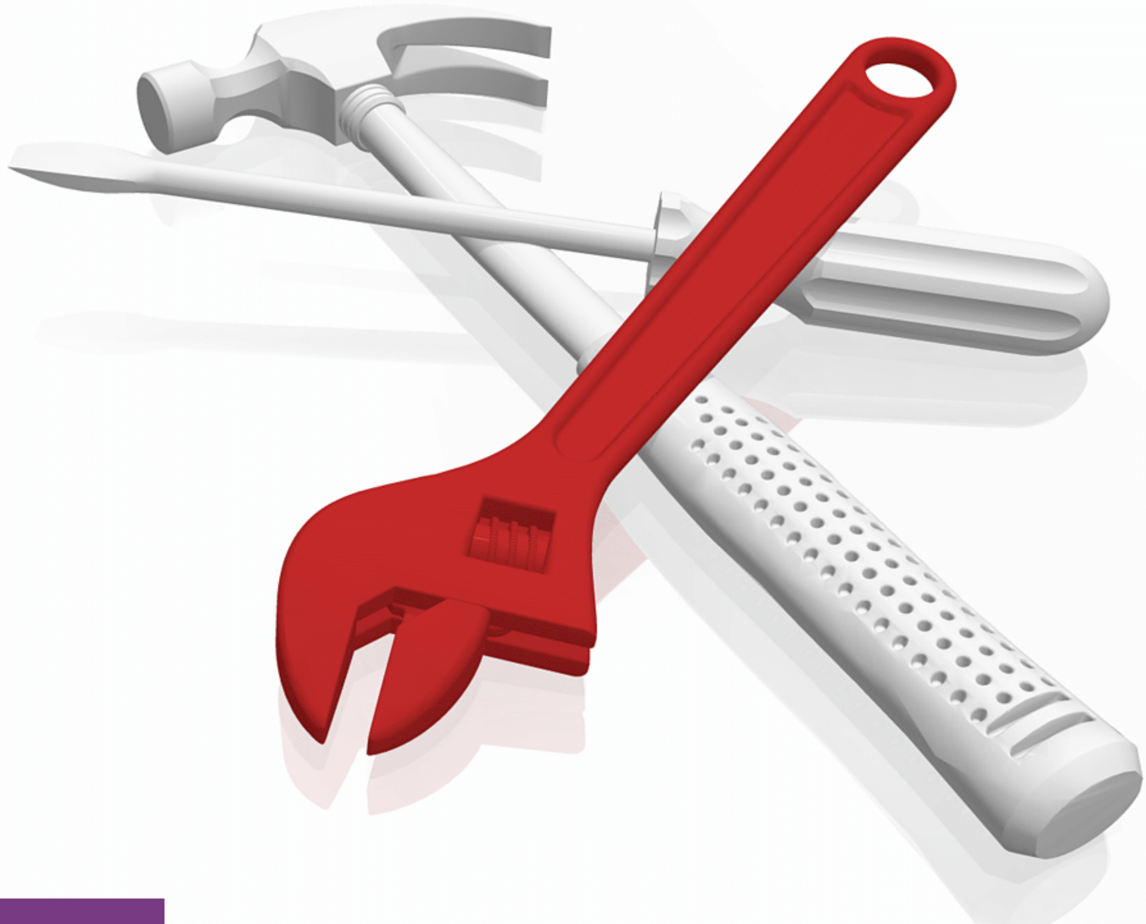
Lydia Parziale

Edi Lopes Alves

Vic Cross

Paul Novak

Mauro Cesar de Souza



IBM Z

IBM

Redbooks

International Technical Support Organization

Virtualization Cookbook for IBM Z Volume 1

July 2021

SG24-8147-02

Note: Before using this information and the product it supports, read the information in “Notices” on page xiii.

Third Edition (July 2021)

This edition applies to IBM z/VM Version 7, Release 2, Red Hat Enterprise Linux version 8, and SUSE Linux Enterprise Server 15 GA, and Ubuntu Server 20.04.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains 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

Preface

This IBM® Redbooks® publication is volume one of five in a series of books entitled *The Virtualization Cookbook for IBM Z*.

The series includes the following volume:

- *The Virtualization Cookbook for IBM z Systems® Volume 1: IBM z/VM® 7.2*, SG24-8147
- *The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux 8.2 Servers*, SG24-8303
- *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354
- *Virtualization Cookbook for IBM Z Volume 5: KVM*, SG24-8463

It is recommended that you start with Volume 1 of this series because the IBM z/VM hypervisor is the foundation (or base “layer”) for installing Linux on IBM Z®. All of the volumes are described next.

Concept of the series

This book series assumes that you are generally familiar with IBM Z technology and terminology. It does not assume an in-depth understanding of z/VM or Linux. It is written for individuals who want to start quickly with z/VM and Linux, and get virtual servers up and running in a short time (days, not weeks or months).

Volume 1 starts with a solution orientation, discusses planning and security, and then, describes z/VM installation methods, configuration, hardening, automation, servicing, networking, optional features, and more.

It adopts a “cookbook-style” format that provides a concise, repeatable set of procedures for installing, configuring, administering, and maintaining z/VM. This volume also includes a chapter on monitoring z/VM and the Linux virtual servers that are hosted.

Volumes 2, 3, and 4 assume that you completed all of the steps that are described in Volume 1. From that common foundation, these volumes describe how to create your own Linux virtual servers on IBM Z hardware under IBM z/VM. The cookbook format continues with installing and customizing Linux.

Volume 5 provides an explanation of the kernel-based virtual machine (KVM) on IBM Z and how it can use the z/Architecture®. It focuses on the planning of the environment and provides installation and configuration definitions that are necessary to build, manage, and monitor a KVM on Z environment. This publication applies to the supported Linux on Z distributions (Red Hat, SUSE, and Ubuntu).

Volumes in this series

This book series consists of the following volumes:

- *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147 introduces the entire concept of the IBM virtualization solution by using z/VM to run Linux servers on an IBM Z system. It also describes the z/VM functions, planning, installation, and configuration of a two-member SSI with z/VM 7.2.

  For Volume 1, you need at least two IBM Z logical partitions (LPARs) with associated resources and z/VM 7.2 installation media.

- *The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux 8.2 Servers*, SG24-8303 describes the installation and customization of RHEL.

  For Volume 2, you need the Red Hat Enterprise Linux Server (RHEL) version 8.2 installation media.

- *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890 describes the installation and customization of SLES.

  For Volume 3, the SUSE Linux Enterprise Server (SLES) version 12 media.

- *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354 describes the installation and customization of SLES.

  For Volume 4, the initial Ubuntu Server 16.04 LTS media plus resources for mirroring.

- *Virtualization Cookbook for IBM Z Volume 5: KVM* describes the kernel-based virtual machine (KVM) on IBM Z and how it can use the z/Architecture.

  For Volume 5, you need KVM distribution installation media.

Main parts of this volume

This volume consists of the following main parts:

- Part 1, “z/VM cloud concepts and planning” on page 11

  In this part, we provide a conceptual overview of how z/VM plays a unique role in virtualization and cloud infrastructures. We also discuss the planning of hardware, software, and networking resources that are needed before you attempt to install z/VM and Linux. We also provide security topics that you should consider before installation.

- Part 2, “Installation, configuration, and service” on page 95

  In this part, we describe how to install and configure z/VM. We discuss live guest relocation and how to service z/VM. We also describe the Centralized Service Management (z/VM CSM) that allows you to manage distinct levels of service for a specific group of z/VM systems.

- Part 3, “System management” on page 303, includes chapters on the following subjects:
  - Implementing live guest relocation (LGR) between SSI members
  - Configuring the Systems Management API (SMAPI)
  - Enabling IBM RACF® as the External Security Manager (ESM)
  - Monitoring z/VM and Linux
  - Describing the Linux system suite of system management daemons, and libraries

Considerations

The “recipes” that are published in this cookbook are meant to teach your brain *how to cook*: that is, suitable techniques, methodologies, safety, and what order things must occur. When you know the *how*, you can more readily understand the *why*. Than, you can choose to tweak and personalize your “menu.”

It is important to realize that the recommendations and examples that are presented in this book and the others in the series are intended to spark thoughtfulness and meaningful awareness of the entire process. In situations where required, things can and should be adapted to suit your specific needs.

Conventions

The conventions that are described in this section are used in this book and others in the series.

Font conventions in this cookbook series

`Monospace and bold`	Commands entered by the user on the command line
`monospace`	Linux file names, directories, and commands
`MONOSPACE CAPITALS`	z/VM files, virtual machine and minidisk names, and commands
`Monospace bold italics`	Values that were used to test this book, such as TCP/IP addresses. This font convention is used to signify that you need to replace the ***`example value`*** with the correct value for your system or enterprise.
`Monospace reversed`	Keys and key combinations that are used to invoke functions or respond.

Command conventions that are used in this cookbook series

- z/VM commands are prefixed with **===>** (three Equal signs (=) followed by a Greater than symbol (>))
- z/VM **`XEDIT`** subcommands are prefixed with **====>** (four Equal signs (=) followed by a Greater than symbol (>)).
- Linux commands that are running as root have a Number sign (#) prefix.
- Linux commands that are running as non-root usually have a Dollar sign ($) prefix.

Operating system releases that are used

The following versions or releases of operating systems were used in the authoring of this book series:

z/VM Version 7.2.0	GA code, September 2020
RHEL Version 8	GA code, May 2019
SLES 15 GA	GA code, May 2020
Ubuntu Server 20.04	GA code, April 2020

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

Lydia Parziale is a Project Leader for the IBM Redbooks® team in Poughkeepsie, New York, with domestic and international experience in technology management, including software development, project leadership, and strategic planning. Her areas of expertise include business development and database management technologies. Lydia is a PMI certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management and has been employed by IBM for over 25 years in various technology areas.

Edi Lopes Alves is a Senior IT Specialist working with IBM Z and LinuxONE for the Global Technical Service team in Brazil. She has more than 25 years of experience working as an IBM z/VM and Linux on IBM Z specialist. Edi has IBM L2 IT Specialist certification, holds a Mathematics degree and a master's degree in e-Business from ESPM Sao Paulo. She currently supports z/VM and Linux on Z for IBM commercial accounts. She supported the z/VM environment and cloud initiatives for Banco do Brasil and IBM Global Accounts (IGA) by supporting IBM Green, IBM Blue® Harmony projects, and z/VM Field Test at Endicott Lab logical partitioned systems. Working across international and diverse teams. Edi has co-authored several IBM Redbooks publications. Edi acquired professional certifications and \ mentored many professionals at different levels of seniority to progress in their careers.

Vic Cross is an IT Solutions Engineer living in Brisbane, Australia. Vic works as part of the IBM Worldwide Z Acceleration Team, helping customers in adopting new technologies, such as Red Hat OpenShift Container Platform on IBM Z and LinuxONE. Before his current role, Vic worked with IBM Systems Lab Services and IBM Systems, where he provided senior design and implementation expertise to IBM Z and IBM LinuxONE projects across Asia-Pacific. He has 30 years of experience in general IT, 25 of which was directly related to the IBM Z and IBM LinuxONE platforms and their antecedents. Vic holds a degree in Computer Science from Queensland University of Technology. He has written and contributed to several IBM Redbooks publications, including *Temenos on IBM LinuxONE Best Practices Guide*, SG24-8462; *Securing Your Cloud: IBM z/VM Security for IBM z Systems and LinuxONE*, SG24-8353; and *Linux on IBM eServer zSeries and S/390®: Virtual Router Redundancy Protocol on VM Guest LANs*, REDP-3657.

Paul Novak is a member of the IBM Advanced Technology Group (formerly known as ATS or Washington Systems Center) z/VM and Linux team in Endicott, New York. Paul has experience in field services, user support, software development, enterprise hosting, enterprise architecture, and customer technical solutioning. His areas of expertise are z/VM, Linux, middleware, containers, automation, and web identity and access management (IAM) solutions. Paul previously worked as IBM technical webmaster on a team that was responsible for consolidation of the world's largest production implementations of IBM WebSphere®, Domino, IBM MQ, and Collaboration Software onto Linux under z/VM. He is a Senior Certified IT Specialist and holds several degrees. Paul is a multi-generation Endicott IBMer and has more than 25 years of Linux and Open Source technology experience.

Mauro Cesar de Souza is z/VM system support specialist working for IBM Global Technology Services® in Brazil, and former IBM® Systems Lab Services engineer, with more than 15 years of experience managing z/VM systems and migrating workloads to the platform, designing, implementing and optimizing solutions. In addition to z/VM, has more than 20 years of experience as Linux administrator, and can program in several languages. He has a degree in Computing Technology from UNICAMP.

With gratitude

The authors of this book extend our thanks to the following people for their contributions and assistance:

Ivan Dobos
Christian Ehrhardt
Frank Heimes
Canonical, Inc.

Chris Mackowski
Jan Stodola
Dan Horak
Red Hat, Inc.

Mike Friesenegger
Ihno Krumreich
Mark Post, Terry Smith, Donald Vosberg

Special thanks

A special thank you to Robert Haimowitz, IBM Garage™ for Systems. Without Bob, these books would not be possible.

Patrik Hysky
Systems Technical Sales Services, Austin

Deyaa Abuelsaad, Z Product Engineering, Poughkeepsie
Alan Altmark, Systems Lab Services, Endicott
Rhette Angus, Shop Z, Atlanta
Fred Bader, Advanced Technology Group, Gaithersburg
Marci Beach, z/VM Development, Endicott
Karen Beale, Z Unit Executive, Advanced Technology Group, Atlanta
John Bernatz, Advanced Technology Group, Lexington
Andreas Bieswanger, IBM Fellow and Z CTO Platform Management, Böblingen
Bill Bitner, z/VM Customer Focus and Care, Endicott
Kay Blake, z/VM Installation, Endicott
Jay Brenneman, Z Systems Integration and Test, Poughkeepsie
Charlie Bryant, IBM z/OS® Function Test, Endicott
Hendrik Brückner, Manager, Linux on Z Development and Service, Böblingen
Eric Carlson, Techline and Shop Z, San Francisco
Tung-Sing Chong, Z Firmware Development, Endicott
Michael Donovan, z/VM® Development and Service, Endicott
Carol Everitt, z/VM Development, Install, Service, and Packaging, Endicott
John Franciscovich, z/VM Development and Service, Endicott
Ingo Franzki, Development and Service, Linux on Z and IBM z/VSE®, Böblingen
Glenda Ford, z/VM Development and Packaging, Endicott
Jacob Gagnon, z/VM Development, Endicott
Leslie Geer, z/VM Development and Release Management, Endicott
Stephanie Gherghe, Linux on IBM Z® and LinuxONE Education, Böblingen
Timothy Greer, z/VM Development, Endicott
Bruce Hayden, Advanced Technology Group, Iowa
John Hollenbeck, z/VM Function Test, Endicott
Ernie Horn, Manager, Global LinuxONE Client Success, Poughkeepsie
Brian Hugenbruch, Z Virtualization and Cloud Security, Endicott
Emily Hugenbruch, z/VM Development, Endicott

Shalawn King, Manager, Advanced Technology Group, Atlanta
Dominik Klein, Linux on Z Performance, Böblingen
Richard Lewis, Advanced Technology Group, Gaithersburg
Jennifer Lynch, Z New Technology Introduction, Burlington
George Madl, Director, z/VM and Related Products, Endicott
Steffen Maier, Linux on Z and LinuxONE, Böblingen
Carl Mayer, Z and LinuxONE Platform Management STSM, Böblingen
Donald McGlynn, z/VM Development, Endicott
Kevin Meeks, z/VM Development, Endicott
Wilhelm Mild, Linux on Z and LinuxONE Container and Mobile Integration, Böblingen
Yanique Moffitt, z/VM Development, Endicott
Matt Mondics, Advanced Technology Group, Cleveland
Kenneth Morse, Advanced Technology Group, Gaithersburg
Kathryn Murtha, Manager, z/VM Development, Endicott
Damian Osisek, z/VM Architectural Design, Endicott
Vasily Papastrat, z/VM Development, Endicott
Pradeep Parameshwaran, Linux on Z and LinuxONE Security, Böblingen
Eberhard Pasch, Linux on Z Architecture and Performance STSM, Böblingen
Keyur Patel, Manager, Z Product Engineering, Poughkeepsie
Patricia Rando, z/VM Development, Endicott
Stefan Reimbold, Linux on Z and LinuxONE Performance, Böblingen
Timothy Reynolds, z/VM Support, Endicott
Clarice Y Robinson, Shop Z, Raleigh
Bob Rogers, Distinguished Engineer, z/VM Development, Endicott
Jon Ruhl, z/VM Development, Endicottt
Tausif A Samad, Shop Z, India
Lucas Sandoval, Techline and Shop Z, Phoenix
Barry Silliman, Advanced Technology Group, Gaithersburg
James Switzer, z/VM Systems Test, Endicott
Jack Sykes, Advanced Technology Group, Boston
Susan Timashenka, Manager, z/VM Development, Endicott
Kara Todd, Director, IBM Z and LinuxONE, Poughkeepsie
Richard Young, Systems Lab Services, Milwaukee
Brian Wade, z/VM Development and Performance, Endicott
IBM

Thanks to Michael MacIsaac for the original inception of this cookbook and for his efforts in continually moving the cookbook forward over the years.

Thanks to many others in IBM Endicott, Böblingen, and Poughkeepsie, and also to the many who answered questions on the `Linux-390` and `IBMVM` list servers.

Thanks to the authors of the previous editions of this book:

The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3, SG24-8147, last updated 27 Jun 2016: Lydia Parziale, Berthold Gunreben, Filipe Miranda, Paul Novak, and Ken Werner.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  ibm.com/redbooks/redbooks.nsf/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

  ibm.com/redbooks/rss.html

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-8147-02
for Virtualization Cookbook: Vol 1: IBM z/VM 7.2
as created or updated on November 11, 2024.

Summary of changes in this book

The following changes were made to this book from the prior publication:

Setup of IBM z/VM:

- The chapter to install a z/VM non-single system image (SSI) logical partition (LPAR) was removed because we encourage you to install SSI if only as a single member cluster. This approach provides the environment to use SSI in the future.
- Naming conventions for virtual network adapters and mini-disks were updated to offer greater clarity and help eliminate confusion.
- The setup of z/VM covers usage of the IBM z/VM Single System Image (VMSSI) feature with a directory management product.
- All of the setup, planning, and service tasks were reworked and reordered into a workflow that is much faster to complete.
- The chapter covering common IBM DirMaint tasks was expanded.
- The section about using emulated DASD (EDEV) was expanded.
- Chapter 10, “DirMaint, RACF-connector, and SMAPI” on page 305 was expanded to include additional detail and a new section on the z/VM Secure Sockets Layer (SSL) server.
- A section about setting up a Link Aggregation Control Protocol (LACP) redundant network configuration on the IBM z13® was added.

Conversion to a volume-based series

- The Red Hat chapters were updated and moved to *The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux Server 8.2,* SG24-8303.
- The SUSE chapters were updated and moved to *The Virtualization Cookbook for IBM z Systems® Volume 3: SUSE Linux Enterprise Server 12,* SG24-8890.
- Information pertaining to Canonical Ubuntu Server was moved to *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354

General

- The chapter to configure an File Transfer Protocol (FTP)/Network File Server (NFS) server on a PC was removed. The availability of an FTP server in the local environment is assumed.

- All information regarding planning and good practices for initial layout, naming, and other wise have been consolidated into Chapter 2, “Planning” on page 33.
- A section covering IBM Dynamic Partition Manager (DPM) was added.

Part 1

z/VM cloud concepts and planning

In this part of the book, we discuss laying the groundwork for using z/VM as an enterprise cloud hosting platform.

This part includes the following chapters:

1

Conceptual overview

This chapter provides a concise introduction to the concept of using IBM LinuxONE or IBM Z as an enterprise-grade private or hybrid cloud infrastructure for Linux workloads.

Although the concepts of virtualization and cloud computing are familiar to many by now, there is information in this section that is likely to be unknown, even to readers who might consider themselves experts. For this reason, it is our recommendation that anyone reading this book should fully read and understand this chapter.

After reading this chapter, you understand why the use of this platform as a fully-integrated cloud solution continues to grow in popularity and how it can surpass virtually everything else from the perspectives of performance, density, and low total cost of ownership.

The conceptual landscape and philosophy that was used in authoring this book are covered first. Next, we cover topics that are critical to understand for success such as terminology, z/VM components, capabilities, and enhancements. Finally, the decisions and assumptions that were made by the authors are reviewed, followed by information about design and usability.

This chapter includes the following topics:

1.1 Basic concepts

IBM z/VM is a highly secure and scalable hypervisor and virtualization technology for cloud infrastructure and for running critical applications. IBM z/VM supports the Linux operating systems on IBM Z and LinuxONE servers.

In this section, we describe the concepts of virtualization and cloud computing.

Virtualization

Virtualization is the ability of a computer system to share resources so that one physical server machine can act as many *virtual servers*. Unlike other hardware platforms, IBM Z and IBM LinuxONE operate on virtualized hardware by default, which results in incredible performance and efficiency.

Virtualization is the way to consolidate the amount of hardware, floor space, and energy consumption in the data center. It also simplifies the procedures to provide reliable, highly available, and seamless serviceability for the virtualization environment.

Virtualization was a hot topic in the overall IT industry for well over a decade and gained even more popularity with the recent explosion of *cloud computing*.

Cloud computing

> *Cloud computing is on-demand access, via the internet, to computing resources — applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more — hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage. The term 'cloud computing' also refers to the technology that makes cloud work. This includes some form of virtualized IT infrastructure — servers, operating system software, networking, and other infrastructure that's abstracted, using special software, so that it can be pooled and divided irrespective of physical hardware boundaries. For example, a single hardware server can be divided into multiple virtual servers." "Cloud computing transforms IT infrastructure into a utility: It lets you 'plug into' infrastructure via the internet, and use computing resources without installing and maintaining them.*
>
> — Courtesy IBM Cloud® Learning Hub

Cloud computing features the following variants:

- Public cloud
- Private cloud
- Hybrid cloud
- Multi-cloud

Because the topic of cloud computing is so expansive, it is impossible to cover with sufficient detail here in this book. Instead, we recommend visiting the IBM Cloud Learning Hub where you can explore each area for more information.

1.2 Why choose this hardware platform, and why z/VM?

For IBM, virtualization and cloud computing are nothing new. Although some users might find it hard to believe, the concepts we all know as virtualization and cloud computing have their roots right here on early generations of this hardware platform. In fact, IBM is widely credited with inventing the system and method for virtualization of a computer system in the late 1960s.

1.2.1 Virtualization and cloud computing originated here

Cloud computing and IBM large systems share a common lineage dating back to the 1950s and 1960s. In those days, enormous mainframe systems were beginning to appear at businesses and universities as the world began to adopt computerization on a large scale.

Early mainframe systems were not the compact and affordable systems they are today. Because of cost and physical size, it was exceptionally rare for any organization to have more than one.

To maximize the utility and return on investment, it became standard practice to provide shared access through physical terminals. Was there a better way? Engineers at IBM's Glendale development laboratory in Endicott, New York began to ponder this question and prototype methods.

In 1971, IBM released a product that is considered the genesis of cloud computing; an operating system called *virtual machine* (VM). For the first time ever, it was possible to have multiple virtual systems, or VMs all co-existing on one physical system. The VM operating system moved the concept of shared access ahead by a massive leap by allowing individual users to effectively have their own personal mainframe.

By the early 1980s, VMs continued to evolve, pioneering ground breaking concepts later to become cloud fundamentals. Automated systems management and self-service functionality for common tasks, such as identity and access management, are just two of many examples.

1.2.2 The ultimate virtualization platform

If someone were to tell you that they recently purchased their "ultimate dream car"; describing a high-performance luxury-sport automobile with a multitude of premium features, you likely assume that it was an expensive purchase with expensive ongoing maintenance costs, and it might not be \ reliable based on all the complexity. What if they then continued on to inform you that:

- The overall total cost of ownership was a fraction of what someone might pay for a basic economy car.
- It is so reliable that empirical data proves a claim by the manufacturer of 50 years mean time to failure.
- It is also can tow the weight of a fully-loaded semi or overseas shipping container.

At this point, you might assume this person to be fantasizing; dreaming about a wish they hope someday might become reality, because this car sounds far too impossible to be real.

Although disappointing, to our knowledge there is no such “ultimate dream car” like the one described here. However, the good news is that the equivalent of this concept as a cloud solution for hosting virtualized Linux workloads does legitimately exist and is available for purchase from IBM today. This solution is the IBM z/VM Enterprise Virtualization Platform, running on IBM Z or IBM LinuxONE servers.

There is overwhelming evidence from independent industry analysts, customer testimonials, and IBM alike to support the argument that IBM z/VM on IBM Z and IBM LinuxONE is the most functionally rich, reliable, and efficient virtualization platform that is available in the world today.

With nearly six decades of continual optimization, enhancement, and refinement of the hardware and z/VM, the result is a premium, high-performance solution: the ultimate virtualization platform.

The following most notable qualities are features:

- Incredible I/O capabilities

  Throughput speeds approach 1 terabyte per second. Unlike other virtualization platforms, z/VM guest systems commonly use real hardware access without another software layer. The platform is referred to by many in the industry as “an I/O monster” because of this superior quality.

- Engineered for availability

  The hardware components in the systems are double, sometimes even triple redundant, with a mean time to failure of over 50 years. If a component problem occurs, often you do not know about it until an IBM service support representative shows up to replace the problem part for you.

- Unsurpassed density

  z/VM can support more virtual servers than any other platform.

- Major cost savings

  With the superior efficiency of the system hardware, the result is an environment that brings tremendous immediate and ongoing cost savings. Reductions in software licensing costs, power usage, cooling, and a fraction of the floor space versus distributed platforms.

- Ultra-low TCO

  When workloads are evaluated from a total cost of ownership (TCO) perspective especially, workloads that are hosted under z/VM on the IBM Z or LinuxONE hardware mainframe have such a dramatic savings benefit to a bottom line that many customers consider this solution to be a competitive secret.

1.2.3 Optimized for Linux

When Linux came to the previous generation of IBM Z in 2000, it was a natural fit to run under z/VM. You can run hundreds of Linux VMs (servers) in the same logical partition (LPAR) under z/VM, restricted only by the amount of available resources. A multi-LPAR mainframe can run thousands of Linux virtual servers, depending on the size and resources that are required.

With a z/VM and Linux infrastructure, you can reduce the time between deciding on the acquisition of new servers and then implementing them because new servers can be easily deployed in a matter of minutes. With this powerful capability, you can launch new products and services without the exhaustive planning for, purchasing, installing, and configuring new hardware and software that can be associated with conventional discrete hardware servers.

Development groups that require test environments that are built and rebuilt rapidly to enable them to efficiently deliver their projects (and handle change management in the process) can also benefit from this unique advantage.

The addition of new companion products, such as IBM Wave for z/VM, now permit many of these tasks to be accomplished with a point and click interface through a web browser.

The following capabilities are several key strengths of IBM Z and IBM LinuxONE and z/VM:

- Their virtualization capabilities are more mature and robust than any other hardware and hypervisor combination.
- The z/VM virtual switch makes networking Linux much simpler and provides automatic redundancy.
- Full volume backup of systems allows for complete disaster recovery when another data center is available.

z/VM is easy to customize at the base installation level because of a relatively small number of configuration files. When z/VM is set up correctly, it can run for months with little required maintenance or administration.

1.2.4 The hidden secret

You might be asking yourself why a solution that fits all of the amazing qualities that were described so far has such little notoriety. Why do we not see the organizations using it touting all of the benefits they get from using it? The simple answer is *competitive advantage*.

Simply put, good business leaders \ go to great lengths to safeguard information about what is sometimes referred to as *the secret sauce*. In reality, this secret can be a reference to anything that makes the products or services they offer superior to any competitors through affordability, quality, or some combination of both. Running their business on this platform is the secret sauce they do not want competitors to know about.

1.2.5 A community of friends

The global z/VM community is known for sharing lessons learned, tips, tools, and advice among each other. You find the listserv and discussion forums that are mentioned in this book to be welcoming and truly useful environments where people are incredibly friendly, helpful, and want to see you succeed.

1.3 The philosophy that was adopted in authoring this book

An important philosophy that was adopted in this book is to keep all solutions simple. Two common expressions that are used are “the KISS method” (Keep It Simple, Stupid) and the quote from Albert Einstein at the start of this chapter: *Everything should be made as simple as possible, but not simpler*. This book will use the latter, with the aim to use the same clear and insightful presentation. The authors thought it was important to help you quickly get a new system up and running, but to also provide insight as to why we chose the options we did.

Many books and papers talk about virtualization and cloud computing concepts today, but they do not tell you *how* to do it. The remainder of this book gives you the “how to” that backs up these marketing concepts.

The setup in this book is considered a well-designed environment by the authors. The topics and directions that are presented also were reviewed by staff in some of the globally recognized subject matter expert and customer practitioner areas in IBM, such as the z/VM development lab, Washington Systems Center (ATS), IBM Garage, and Global Technology Services.

Implementing this environment results in a robust virtualized cloud environment that can serve as a base for many upcoming new tasks and workloads.

1.4 A high-level overview of components and terminology

We provide an overview of the components and technology because understanding what you are working with is critical to successful planning and deployment.

1.4.1 Hardware

Consider the following hardware concepts:

- Central processor complex

  Depending on the context of a discussion, the terms *processor* and *CPU* can refer to the complete system box, or one of the central processors (CPUs) within the system box.

  Although the meaning might be clear from the context of a discussion, even IT professionals must clarify their intended meaning when they use the terms *processor* or *CPU*.

  IBM uses the term *central processor complex* (CPC) to refer to the physical collection of hardware for IBM Z that includes main storage or memory, one or more central processors, timers, channels, interconnects, peripherals, cooling apparatus, and more.

  Professionals typically use the term *system* to indicate the hardware box, a complete hardware environment (with I/O devices), or an operating environment (with software), depending on the context. They typically use *processor* to mean a single processor (central processor (CP/GP) or Integrated Facility for Linux (IFL)) within the system.

  > **Note:** Despite the outward similarities in appearance between the IBM Z and IBM LinuxONE systems, they are indeed two distinctly different systems. As such, it is not technically accurate or correct to refer to a LinuxONE system frame as a CPC. The correct terms to use for LinuxONE are *server* or *system*.

Figure 1-1 shows an IBM Z 15 CPC on the left and a IBM LinuxONE III server on the right. More distinctions between the two system types are covered in greater detail later on in this chapter.

> **Important:** You might hear some system programmers use the antiquated term *central electronic complex*, which is abbreviated as CEC and sometimes even pronounced "k'ehk" (as if an actual word).
>
> CEC (nor any pronounced variant of CEC) is not sanctioned by IBM for use. It is considered to be highly offensive in some languages and cultures, and should not be used.
>
> In addition, today's modern CPC evolved far past the point of a simple electronic complex. Referring to a modern Z CPC as a CEC is analogous to calling a modern state-of-the-art smart refrigerator an "icebox".

Figure 1-1 IBM Z 15 model T02 (left) and IBM LinuxONE III model LT2 (right)

- Processing units

  Briefly, all of the IBM z/Architecture processors within a system are processing units (PUs). When IBM delivers the system, the PUs are characterized as different types based on the workloads they process.

  For IBM Z, these PUs can be General Processors (GP or CP), Integrated Facility for Linux (IFL), Integrated Coupling Facility (ICF) for IBM Parallel Sysplex® configurations, and so on.

  For LinuxONE, the system is equipped with only Integrated Facility for Linux (IFL) processing units.

- Logical partitions

  From the perspective of system use and management, LPARs are equivalent to separate physical systems. Each LPAR has its own operating system, which can be any mainframe operating system. LPARs optionally also can be configured to share I/O devices, but this decision is a local decision that is made during hardware planning.

The system administrator can assign one or more system processors for the exclusive use of an LPAR. Alternately, the administrator can allow all processors to be used on several or all LPARs. Here, the system control functions, which are often known as *microcode* or *firmware,* provide a dispatcher to share the processors among the selected LPARs. The administrator can specify a maximum number of concurrent processors to run in each LPAR. The administrator can also assign *weights* to LPARs; for example, specifying that LPAR1 receives twice as much processor time as LPAR2 by adjusting the weighted value.

The operating system in each LPAR undergoes a separate IPL, has its own copy of its operating system, has its own operator console (if needed), and so on. If the system in one LPAR crashes, other LPARs are not affected.

- Real storage/central storage (memory)

  Both of these terms refer to *main system memory*, which is also frequently called *storage*. Anyone who has experience with mainframes or mobile and tablet devices understands the notion of storage being a reference to memory. For others, it might take time to become accustomed to this concept.

- Open Systems Adapter card (OSA)

  This card is the networking adapter that is used in IBM z/Architecture systems.

- Direct access storage device (DASD)

  IBM disk storage subsystems, such as IBM System Storage DS8900F and TS7770, are designed to match the mission-critical capabilities of IBM Z and LinuxONE. They add next-level performance, security, and resilience for mission-critical mainframe storage workloads across hybrid multi-cloud deployments.

  The DS8900F, as is the case with previous iterations in the IBM DS8000® family of storage subsystems, provides emulated IBM machine type 3390 disk drives of varying model types. Common models are 09 (3390-09), 27 (3390-27), and 54 (3390-54).

  Because of their extremely small size, model 03 (3390-03) DASD do not have the popularity they once did. The 3390-03 DASDs are also no longer supported for use in the installation of z/VM. As such, the other models of 3390 DASDs that are still commonly used and supported with current installations are covered in much more detail throughout this book.

- Channel-to-channel Adapter (CTCA)

  Inter-System Facility for Communications (ISFC) for inter-cluster communication for LGR. ISFC uses channel-to-channel (CTC) devices.

- Input/Output Configuration Data Set (IOCDS)

  An I/O configuration data set (IOCDS) contains information that is used to define the I/O configuration to the processor complex's channel subsystem.

1.4.2 Software

The software components are described in this section.

z/VM Enterprise Virtualization Platform

Within the physical system, z/VM is installed as the operating system for an LPAR. z/VM allows the sharing of the physical resources that are accessible to that LPAR. Physical resources can include disk (DASD), memory (sometimes called *storage*), network adapters (OSA cards), and PUs (CPs or IFLs).

z/VM features the following key aspects:

- Control program

  Resources that are available to each LPAR that is running z/VM are managed by the z/VM hypervisor, which is known as the control program (CP). When the user logs on to z/VM, the hypervisor creates a VM, which can run one of many different operating systems. The two operating systems that are described in this book are Conversational Monitor System (CMS) and Linux.

- Conversational Monitor System

  CMS can be thought of as a z/VM *shell* because the outward function is similar in concept to that of the Bourne shell in Linux.

  > **Note:** Multics, which inspired UNIX, shares a common background with the z/VM Conversational Monitor System and some components of the CP.

- Virtual Machine

  From a z/VM perspective, the following terms all refer to the same thing; that is, a VM definition:

  – User
  – ID
  – Identity
  – Guest
  – VM
  – Virtual Server

  A VM definition is a unique entry inside of the z/VM User Directory, which consists of a collection of parameters and defined resources that begin with the term `USER` or `IDENTITY`.

  > **Note:** For clarity and consistency, *virtual server* is the preferred term that is used in this book wherever possible to refer to a Linux VM or server.
  >
  > For more information, a description of VM types, and the `USER` and `IDENTITY` statements, see *z/VM Getting Started with Linux on System z*®, SC24-6194.

- VM Single System Image feature (VMSSI)

  Beginning back with z/VM 6.2, the VMSSI feature made it so that a grouping of up to four LPARs running z/VM can be managed as an interconnected cluster. Each member LPAR can exist on the same physical system, or, any physical systems that are close enough for channel-to-channel inter-connectivity. Based on a statement of direction from z/VM development, in the near-term it is be possible to create VMSSI clusters with up to eight members.

- VM Single System Image Live Guest Relocation (LGR)

  It is possible for actively running Linux virtual servers to be relocated from one z/VM system to another within the same VMSSI cluster. The VMSSI feature provides this mobility through what is called *live guest relocation* (LGR).

  You might need to relocate a running VM for several reasons, for example, workload rebalancing, software configuration management, or hardware maintenance. Before you relocate a guest, you must understand the architectural, disk, memory, and networking requirements. Within this book, hints are provided to help with the installation of the VMSSI feature, and tips are provided to relocate a Linux guest.

1.4.3 z/VM capabilities and enhancements by version and release

An incredible number of significant enhancements and new functions were added to z/VM over the last two versions of the product.

The following levels are supported at the time of this writing:

- Version 6, Release 4. (6.4)
- Version 7, Release 1. (7.1)
- Version 7, Release 2. (7.2)

The following brief summary describes what is new or enhanced by version and release.

z/VM 7.2

Debuting September 2020, z/VM 7.2 provides organizations with the premier hypervisor for hosting enterprise-class virtual servers to use the IBM Z and IBM LinuxONE advantages in scalability, performance, high availability, and security.

The objective of z/VM 7.2 is to enhance the capabilities that support digital transformation, specifically those capabilities that are associated with scalability and efficiency. z/VM 7.2 is designed to enable the deployment of up to thousands of Linux servers on a single IBM Z or LinuxONE server

Although cloud computing became the standard use model for IT services, an IT infrastructure continues to be the foundation for every IT service. Realizing the benefits of cloud computing requires an infrastructure that delivers availability, reliability, security, and performance, while also providing strong virtualization technology, such as z/VM.

Virtualization is fundamental to delivering infrastructure as a service (IaaS), which is the basic building block for cloud. IBM continues to invest in z/VM technology to provide leading-edge virtualization capabilities for enterprises that use IBM Z and IBM LinuxONE platforms, while evolving to meet the needs of IT organizations to deliver the foundation for user satisfaction. This progress can meet the needs of IT organizations to deliver the foundation for user satisfaction on traditional and modern workloads, including:

- New container implementations using Red Hat OpenShift Container Platform and IBM Cloud Paks for Linux on IBM Z and IBM LinuxONE
- Traditional monolithic workloads

Because the list of enhancements continues to grow with each release, an abbreviated listing is presented here. For more information about the full and complete list of enhancements, see this web page.

With z/VM 7.2, IBM continues to deliver the following product enhancements to its z/VM advanced virtualization technology on IBM Z and LinuxONE servers:

- VM Centralized Service Management (VMCSM) for non-VMSSI environments
- Multiple Subchannel Set (MSS) MT-PPRC support for the IBM GDPS® environment
- Support for the z/VM ADJUNCT environment
- Planned near-term future support for:
  - Eight-member VMSSI clusters
  - z/XC Architecture support
  - Cylinder 001-END Minidisk HyperPAV

- System default changes:
  - TDISK clearing default changed to Enabled. The default can be turned off by using the FEATURES DISABLE CLEAR_TDISK configuration statement. This change causes any residual data that might be otherwise left on a temporary disk after use to be purged by default, which enables z/VM to be more in line with modern security guidelines.
  - z/VM Directory Maintenance (DirMaint) NEEDPASS default value changed to No. Based on customer feedback and guest service machine requirements, a need no longer exists for an extra layer of authentication. This change enables users and service machines to enter DirMaint commands without supplying passwords. It also eliminates the need to update specific configuration options.
  - z/VM DirMaint default DVHWAIT BATCH and CLUSTER INTERVAL values were updated to improve DirMaint's overall processing time in response to directory change requests.
  - The default unparking model changed from LARGE to MEDIUM. This change reduces the tendency to use vertical-low (VL) logical cores and might reduce the overhead that is induced in the Processor Resource/Systems Manager (PR/SM) hypervisor and improve the use of processor cache.
  - System Recovery Boost was enabled by default, which allows z/VM to automatically use boosting sub-capacity processors to full capacity for a limited duration during start and shutdown when running on the IBM z15™.
  - The PAGING63 IPL parameter and associated external interfaces were removed. z/VM paging subsystem improvements include support for IBM High Performance Fibre Connection (FICON®), IBM HyperParallel Access Volumes (HyperPAV), encryption, and IBM Extended Address Volume (EAV), which are not available when the PAGING63 IPL parameter is specified.
  - The Environmental Record Editing and Printing Program (EREP) licensed program is no longer preinstalled with z/VM. Instead, EREP functional code is preinstalled and delivered as part of the z/VM 7.2 product and serviced through the Control Program (CP) component, simplifying the process for applying EREP service.

z/VM 7.1

IBM enters into a new era for delivering enhancements to z/VM advanced virtualization technology with z/VM Version 7.1, which debuted September 2018. Consider the following points:

- With the adoption of the z/VM Continuous Delivery model, IBM offers z/VM V7 customers timely introduction of new functions in the service stream and a two-year release delivery model that helps customers predictably manage the lifecycle of their z/VM systems.
- To support continuous deployment of the new z/VM functions, z/VM V7.1 includes the Single System Image (VMSSI) function as part of the z/VM V7.1 base at no extra cost.

The following enhancements that were delivered in Version 7 Release 1 are significant:

- New z/VM function, as Small Programming Enhancements (SPEs), are delivered in the service stream of the current Version 7 release. When a new release is introduced, SPEs are delivered on that release that goes forward and, with a few exceptions, the earlier release delivers corrective service only and no new function. When z/VM V7.1 becomes available, licensed users of z/VM V6.4 receive only corrective service.

- Beginning with Version 7.1, IBM delivers z/VM releases on a fixed, 24-month cycle. These releases are a cumulative roll-up of:
  - Previously-released SPEs
  - New function that is too disruptive or pervasive to ship in the z/VM service stream
  - Fixes that were shipped in the service stream of the earlier release
- VMSSI is included in the base of z/VM V7.1 at no extra cost. Previously, it was a priced feature of z/VM V6. Integrating and making VMSSI available at no charge is intended to help more customers reduce or shorten planned outages of their Linux workloads as they adopt the z/VM Continuous Delivery model for their z/VM systems.

  VMSSI includes Live Guest Relocation and single system maintenance to give customers a mechanism to host Linux virtual server images without suffering interruptions as they apply updates to their z/VM system.
- Dump processing is enhanced to reduce the time required to create, process, and transmit data from SNAPDUMP and hard Abend dumps. By default, these dumps are considerably smaller, and require less space in the system SPOOL and CMS file system. The increased efficiency of dump processing can help save time and resources, and remove an inhibitor to the deployment of z/VM configurations with large amounts of memory. The PTF for APAR VM66176 further reduces the time that is required to create a SNAPDUMP or HARD Abend dump.
- Dynamic external security manager (ESM) protection support for the CPACCESS, CPTYPE, and CPVLOAD commands enables these commands to use the current dynamic command protection setting of the **LINK** command when validating the required LINK authorizations. It also ensures the ESM is called only when it is configured to handle LINK authorization requests.
- QUERY BYUSER support for class B users provides privilege class B users the ability to issue the **QUERY BYUSER** command for other users, which is similar to the function that is granted by privilege class E.
- When an ESM is present, programs can use the ESM for all SMAPI authorization decisions at the same granularity that is used with the SMAPI authorization mechanism. The ESM logs the decision (or not) that is based on its active policy, without SMAPI knowledge or intervention.
- The z/VM Cloud Connector is a development toolkit that manages z/VM host and VMs. It provides a set of RESTful APIs to operate z/VM resources. Upper layer cloud management solutions can use these RESTful APIs directly to manage z/VM. For more information, see this web page.
- Dynamic Memory Downgrade Planned

  The flexibility to reassign (add and remove) system resources is critical to z/VM customers. Today's workloads are no longer static. Memory configuration requirements for z/VM images are highly variable because of the nature of constant changing demands within guest workloads. z/VM images can regularly require extra memory to handle short term increases in memory demands. customers require a mechanism to remove this added memory later after workload memory demands diminish. This action must be done without requiring an IPL.
- Elliptic Curve Cryptography (ECC)

  With the PTF for APAR PI99184, the z/VM TLS/SSL server is enhanced to improve security through enabling ECC cipher suites. ECC provides a faster, more secure mechanism for asymmetric encryption than standard RSA or DSS algorithms.

- With the PTF for APAR VM66174, the RSCS server is enhanced to provide a means to query the service level of each part that is included within the RSCS LOADLIB. A new RSCS query parameter is provided that returns the highest level PTF that is applied to each part within the running RSCS server. This ability eliminates ambiguity on whether service was applied.
- IBM has a long history of working with customers to deliver capabilities to improve z/VM. IBM takes this interaction to a new level because z/VM customers might be enlisted as “Sponsor Users” to advise IBM throughout the design process for many z/VM development projects. These customers might also test early versions of the new support before its delivery to the marketplace to ensure their expectations are met or exceeded. IBM finds the Sponsor User relationship to be beneficial and is soliciting more z/VM customers to become involved in this process.

  IBM publishes information about many of its z/VM development projects to help users decide if they want to volunteer as Sponsor Users and also to help the community at large plan for the introduction of new z/VM function.

  This new level of communication between IBM and the z/VM user community facilitates discussion regarding implications of the planned support, such as operational incompatibilities, changes to system behavior, and software vendor impacts. For more information about updated plans, see this web page

z/VM 6.4

IBM z/VM V6.4 (November 2016 - March 2021) helps to extend the business value of IBM z Systems and IBM LinuxONE technology across the enterprise. z/VM V6.4 virtualization technology is designed to run hundreds to thousands of Linux servers on a single IBM z Systems or LinuxONE server with the highest degrees of efficiency and elasticity.

A fundamental strength of z/VM is the ability for VMs to share system resources with high levels of resource usage. z/VM V6.4 provides even greater levels of extreme scalability, security, and efficiency to create opportunities for cost savings, while providing a robust foundation for cognitive computing on z Systems and LinuxONE servers

z/VM 6.4 delivers the following key enhancements:

- Support for up to 2 TB of memory that enables:
  - Higher levels of workload consolidation
  - Considerable growth in memory-intensive applications
  - Superior levels of elasticity for workload spikes
- Increased efficiency with HyperPAV paging that uses IBM DS8000® features to increase the bandwidth for paging and allows for more efficient memory management of over-committed workloads
- Easier migration with enhanced upgrade-in-place infrastructure that provides an improved migration path from previous z/VM releases
- Improved operations with ease-of-use enhancements requested by customers that include:
  - Querying service applied to the running hypervisor
  - Providing environment variables to allow programming automation, based on systems characteristics and customer settings
  - Improved query capabilities for system shutdown

- Improved SCSI support for guest attachment of disk and other peripherals, and hypervisor attachment of disk drives to z Systems and LinuxONE systems to:
  - Increase efficiency by allowing an IBM FlashSystem® to be attached to z/VM for system use without the need for an IBM System Storage SAN Volume Controller
  - Enable ease of use with enhanced management for SCSI devices to provide information needed about device configuration characteristics
  - Allow concurrent code loads on the SVC, and devices incorporating SAN Volume Controller technology, without acquiescing EDEVICE I/O
- Increased scalability by using Guest Enhanced DAT to allow VMs to use large (1 MB) pages, which decreases the memory and overhead that is required to perform address translation
- Integration of new CMS Pipelines functions, which were not previously incorporated within z/VM, that allows a much more inclusive set of tools for application developers
- Availability of IBM Wave for z/VM as an optional, priced feature

z/VM 6.3

z/VM 6.3 (July 2013 - December 2017) extends IBM Z and IBM LinuxONE virtualization platform to help you reshape and derive more value from your systems. It was designed to offer the following benefits:

- Improved economies of scale with z/VM support for 1 TB of real memory:
  - Better performance for larger VMs

    Quadruples memory scalability while continuing to maintain greater than 90% resource utilization
  - More vertical scalability to help reduce LPAR sprawl

    Considerably more VMs can be consolidated into a single LPAR, depending on workload characteristics
  - Reduced administrative expense through managing a smaller number of large-capacity z/VM host servers
- Improved performance with z/VM HiperDispatch.
- More efficient use of CPU hardware resources for dispatched work.
- IBM adopted OpenStack as part of its cloud strategy. IBM is making contributions to the OpenStack project that are designed to enable z/VM 6.3 to be the first IBM System z® operating environment to be managed by these open cloud architecture-based interfaces.
- Simplified migration to z/VM V6.3 with *upgrade in place*, which reduces the effect of an upgrade on active workloads.
- Highly secure industry-standard support that is required for banking and financial-industry applications.
- Support for the new IBM zEnterprise® EC12 (zEC12) and IBM zEnterprise BC12 (zBC12) servers.
- The use of expanded memory is now deprecated.

z/VM 6.2

z/VM 6.2 (December 2011 - June 2017) continues to help customers extend business value across the enterprise by integrating applications and data while providing high levels of availability, security, and operational ease.

This release implements multi-system virtualization of up to four z/VM systems. This technology extends z/VM virtualization to a greater level, which enables members of the cluster to share resources and synchronize. This technology also gives the appearance of being a single system image (VMSSI).

With the IBM z/VM Single System Image (VMSSI) feature, a running Linux VM can be relocated non-disruptively from one member system to any other member, a process that is known as *live guest relocation* (LGR). LGR provides application continuity across planned z/VM and hardware outages.

Members of a cluster are part of the same Inter-System Facility for Communications (ISFC) collection, and use ISFC channel connections to communicate. Multiple channel-to-channel devices provide a greater capability for data to flow between members. All members of a cluster share DASD for VMs and selected z/VM data. Sharing minidisks between members improves the integrity and performance of the system and provides a foundation for LGR.

Members of a z/VM VMSSI cluster are managed, serviced, and administered as one system. Resources, including the user directory, minidisks, spool files, and network devices that are used by the control program (CP) and VMs are shared among all members. Sharing of resources helps give Linux guests access to the same devices and networks, regardless of which member they are logged on to or where they are relocated.

Each member of a z/VM VMSSI cluster can communicate with other active members. When a z/VM system is configured as a member of a cluster, it automatically *joins* the other members during system start. Coordination of members that are joining and leaving the cluster, maintaining a common view of member and resource states, and negotiating access to shared cluster resources are all accomplished in a seamless fashion. This coordination allows Linux guests to be relocated between members during planned outages.

Linux guests can now be moved from one member to another during most planned outages (service upgrades) without interruption. This capability allows the Linux application continuous run time during planned outages, and therefore allows the application to experience no downtime.

To use the functions that define and maintain an VMSSI cluster, the IBM VMSSI feature must be licensed and enabled. Servicing in an VMSSI cluster is simplified by using a single service stream for all members. Sharing service resources allows service to be rolled out to each member of the cluster on individual schedules, which avoids an outage for the entire cluster. This capability allows uninterrupted Linux guest availability because the Linux guest can be relocated to a different member before a planned outage.

1.5 Choices and decisions for this book

When we were deciding on installing, maintaining, and provisioning Linux virtual servers under z/VM, we made many basic decisions. The following choices and assumptions were made in this book:

- Use of systems management products: Because this book is designed for you to learn the basics, whenever systems management products are used, we highlight the role that is played by the add-on product.

  We also describe what must be performed in the absence of the add-on product, or provide a reference as to where you can find the information to do so.

- The authors assume that you are going to use a single system image (VMSSI) environment for continuous operation during service. To simplify operation, prevent administration errors, and provide synchronization of directory data among VMSSI members, the use of a directory maintenance product, such as IBM *DirMaint* or CA Technologies *VM:Direct* is a prerequisite. This book describes DirMaint installation and uses it for all directory management tasks.
- Suitable system security and hardening are considered from the start. Directions are included to install and configure both the VM SSL server and an External Security Manager (ESM).

> **Attention:** It is bad practice to run a z/VM system without an ESM in place to correctly encrypt passwords and ensure sufficient controls on commands and system functions.
>
> It is widely regarded as good practice for any non-production systems (such as those systems for development and testing) to be administered to a production-level standard. As such, z/VM always must be configured to use an ESM, regardless of any functional readiness or status labels.

- Use of a Shared File System (SFS) file pool inside of z/VM acts as a central repository for the Linux kernel and master copies of parameter and configuration files.
- A writable Linux `/usr/` file system that is unique to each Linux guest is used. Specific solutions create an environment that shares the `/usr/` file system across all Linux guests as read-only. This approach often makes the solution more complex by requiring more planning, especially when adding or updating software that is used by the virtual servers. A read/write `/usr/` file system on the virtual servers is chosen to keep things as simple as possible.
- Conventional 3390 IBM extended count key data (ECKD) DASD, fixed-block architecture (FBA) disks that are accessed with Small Computer System Interface (SCSI) over Fibre Channel Protocol (FCP), and emulated DASD (EDEV) are all described. However, the use of conventional 3390 DASD is still considered to be the fastest and simplest choice.
- To avoid unnecessary risk, the authors chose not to cover cloning. As an alternative to cloning, automated installation or imaging is now considered the preferred practice and is covered in this publication.

> **Important:** The practice of cloning is no longer considered a good practice and is deprecated. Changes in modern Linux that result from the adoption of systemd as a central service bus render the practice of cloning systems to be risky and uncertain.

1.6 Single system image design

With the introduction of z/VM 6.2 in December 2011, the architecture of Linux solutions on this platform changed dramatically. It is true that Cross Systems Extensions (CSE) allowed for a type of clustering environment for Linux on z before z/VM 6.2. However, CSE was not widely used nor was the architecture completely enabled for clusters.

z/VM 6.2 introduced VMSSI with LGR and brought about major changes. No longer is it true that a z/VM system is the most important “object” in the hierarchy. With z/VM 6.1 and earlier, the system identifier of each z/VM system was the most important. With z/VM 6.2 and later, the VMSSI name is the highest level identifier.

A block diagram of a four member VMSSI, with default volume labels, is shown in Figure 1-2. The recommend scenario for full continuous availability is a four-member cluster with two members on two different CPCs. Figure 1-2 uses this layout.

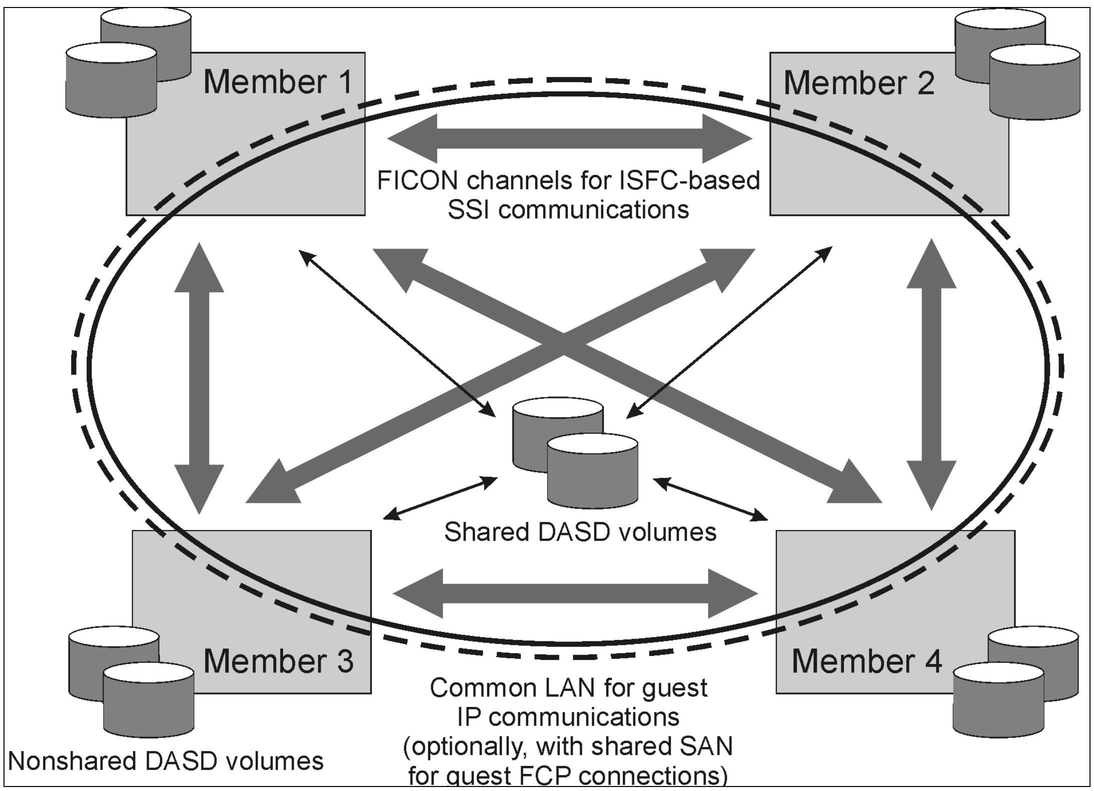


Figure 1-2 A four-member VMSSI cluster architectural diagram

Four z/VM systems and four system identifiers are in this cluster. However, only one VMSSI name is in this cluster. In this book, a two member VMSSI that is installed onto one CPC is described, as shown in Figure 1-3.

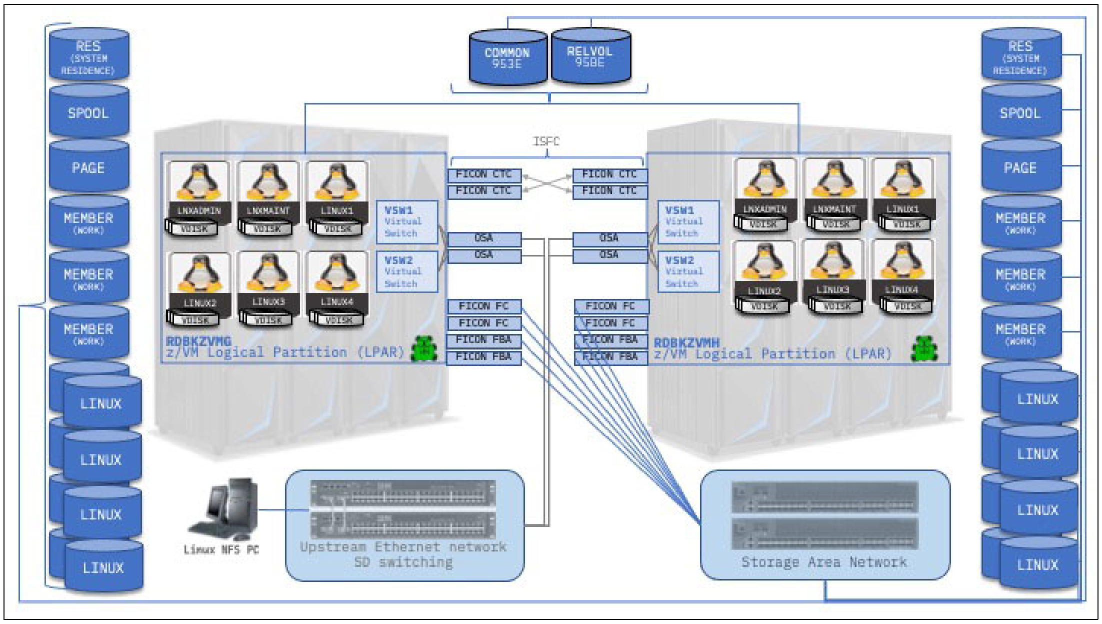


Figure 1-3 The z/VM environment used in this book

1.7 Infrastructure design

To install and configure z/VM, and to install, configure, and provision virtual servers, a specific infrastructure design must be in place. A CPC with associated resources and the z/VM operating system define much of this infrastructure. z/VM includes many predefined virtual servers.

The VMs that are described in this book have the following functions:

LNXMAINT A VM that is used to own and manage the Shared File System (SFS) file pool to be used by Linux virtual servers.

LNXADMIN The Linux system administration server that owns data in the SFS file pool, controls access control list (ACL) entries on SFS, and performs other system administrative functions. This identity can be logged on to all VMSSI members at the same time.

LINUX1 - LINUX4 Four sample worker virtual servers.

In addition to the two LPARs, two other machines are shown:

External FTP server A Linux box that is used for the initial installation of z/VM and each distribution

Workstation machine

A workstation from where all of the work is performed

1.8 Usability tests that are performed

During the years of writing of this book, many usability tests were conducted. The participants had various skills, but none had solid Linux and z/VM system administration skills. By the end of two days, most participants created their first Linux virtual server. You might be able to complete the steps in the book in two to three solid days of work, if all goes well and you work hard.

1.9 Critical differences of LOGOFF versus DISCONNECT

This topic might seem like a simple topic that does not require mentioning, but a critical difference exists between these two commands that you must understand:

- **LOGOFF**

  If you log off, the session is ended, but not in the way you might expect.

  When the ID or VM is running CMS, issuing the **LOGOFF** command is analogous to cleanly shutting down and powering off a desktop or notebook computer.

  When the ID or VM is running Linux, issuing the **LOGOFF** command is analogous to pulling the electrical cord out of the outlet from a desktop computer, or ripping the battery out of a notebook that is not plugged in. The Linux file system journal is then left in an inconsistent state and the risk of file system corruption or loss of data is introduced.

  Consider the following information for the use of the **LOGOFF** command:

  – Use **LOGOFF** only with system administration VMs, such as `MAINT720`, `MAINT`, `TCPMAINT`, and `LGLOPR`.

  – Never use **LOGOFF** with a VSM that is providing a service or running automation, such as RACFVM, DIRMAINT, and TCP/IP.

  – NEVER use **LOGOFF** with a Linux virtual server while Linux is still running.

  – **LOGOFF** destroys ALL running processes for that VM or server and ends any connections.

- **DISCONNECT**

  If you run the **DISCONNECT** command, your session remains where it is and is resumed when you next run the login process.

  It is analogous to:

  – Turning off the monitor of a desktop computer

  – Using a Linux terminal with a utility, such as `screen`, `tmux`, or `byobu`, that permits detachment and re-connection of a running session.

  – Closing the lid of a notebook that is set to keep running and not suspend or hibernate

  Always **DISCONNECT** from z/VM service machines, such as `TCPIP, RACF, DIRMAINT` and virtual servers that are running Linux.

1.10 Summary of Linux and z/VM similarities

Although Linux and z/VM differ in many ways, they feature similar concepts, functions, and commands, as listed in Table 1-1.

Table 1-1 Conceptual similarities between Linux and z/VM

Linux	z/VM (CP and CMS)
Boot	IPL (initial program load)
File system / directory	Minidisk Shared File System (SFS)
File system mount	Disk access mode
Kernel	Control program (CP) Nucleus
Memory (RAM)	Storage
$HOME	Disk access mode “A” (A/K/A “A-DISK”)
~/.profile	PROFILE EXEC A
Script (executable file .sh, .ksh, .pl, and similar)	EXEC or REXX
Shell	Conversational Monitor System (CMS)
User registry (/etc/passwd or /etc/shadow)	User directory file
vi, vim, emacs, nano, pico, or similar	XEDIT

Table 1-2 lists similar commands and functions.

Table 1-2 Similar common commands and functions between Linux and z/VM

Linux command	z/VM CP or CMS command
`df`	QUERY DISK QUERY ACCESSED
`dir, ls`	LISTFILE
`ls -alp`	FILELIST
`man, apropos`	HELP
`free`	QUERY VIRT STORAGE
`uname -a`	QUERY CPLEVEL
`who`	QUERY NAMES
`uptime`	QUERY CPLEVEL *(IPL at .....)*

2

Planning

This chapter covers the planning of hardware, software, and networking resources that you must complete before you attempt to install z/VM and Linux.

First, planning for the mode in which you operate your IBM Z or LinuxONE system, the type of z/VM installation you use, and comparing z/VM running in a VMSSI cluster versus running standalone z/VM systems is described.

Then, we discuss of the *bill of materials,* which is a listing of all of the necessary resources. Next, we describes standardized conventions and good practices that should be adopted for labeling, configuring, and using system resources.

Finally, a planning resource worksheet is presented to document all of the values that were obtained during planning.

The planning resource worksheet covers the following resources:

- z/VM resources
- Linux resources
- Linux virtual machines (VMs)

The previous edition of this publication exclusively covered z/VM with the VMSSI feature enabled. This decision was made because of the increasing popularity of the VMSSI feature, which was furthered by the inclusion of VMSSI at no extra charge in the z/VM base as of Version 7.1.

However, non-VMSSI installations are still in use, mostly at installations where ECKD DASD, as described in Part 1.4.1, "Hardware" on page 18, is not present. When z/VM is installed to SCSI-over-FCP disk, the use of VMSSI is not possible.

In addition, a new service mode was introduced with z/VM 7.2. Called *VM Centralized Service Management* (VMCSM). This capability allows a single non-VMSSI system to generate service updates for other non-VMSSI systems.

If you are just getting started with z/VM and were not planning to use the VMSSI feature immediately, we still encourage you to install z/VM as a VMSSI cluster with only one member node to facilitate expansion in the future. Adding nodes to an existing cluster is much easier than having to start from square one.

This chapter includes the following topics:

2.1 Hardware operation and interface mode

If you are new to Linux on z/Architecture hardware, you must decide to which interface mode your system will operate in:

- PR/SM Logical Partition (LPAR)
- Dynamic Partition Manager (DPM)

This section describes the two options in an effort to help you make an informed decision.

If you purchased the IBM Dynamic Partition Manager (DPM) feature and do not plan to run other IBM Z operating systems, you likely want to consider the use of DPM to operate your system and manage the partitions you create.

Regardless of which method you choose, remember that IFLs are *never* assigned to an PR/SM LPAR or DPM partition. Rather, the weights are set to move the entitlements to the LPARs or partitions that are deemed to be higher priority. For more information, see “z/VM configuration and performance information from Dr. Brian Wade:” on page 476.

2.1.1 Processor Resource/Systems Manager

Processor Resource/Systems Manager (PR/SM) enables logical partitioning of the central processor complex (CPC) in IBM Z and IBM LinuxONE servers. Each LPAR runs its own operating system, such as z/VM, and include resources that are dedicated or shared among multiple LPARs. It allows the definition and control of the logical partitions and their resources, including the following examples:

- Processors
- Memory
- Channel paths

The IBM Z virtualization landscape is unique by providing native support for two levels of virtualization: The first level is used by PR/SM to enable logic partitioning and the second level is used by z/VM to enable the efficient execution of virtual machines.

z/VM is a specialized operating system entirely dedicated to the execution of virtual machines which makes it a type 1 hypervisor (see Figure 2-1).

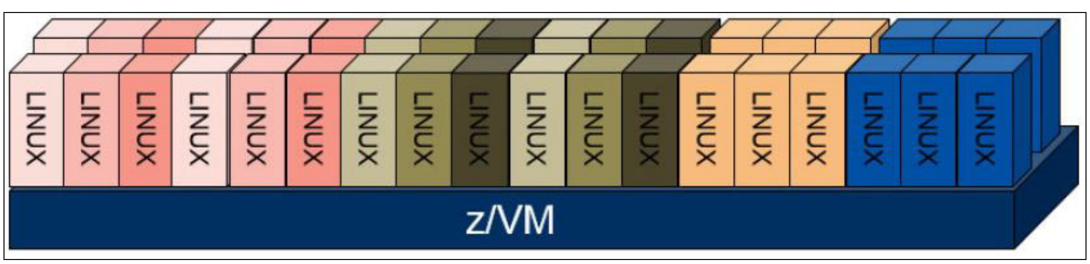


Figure 2-1 Hardware and software hypervisors

2.1.2 IBM Dynamic Partition Manager

IBM Dynamic Partition Manager (DPM) is a new administrative mode that was introduced to LinuxONE servers. A system can be configured in DPM mode or PR/SM mode. DPM provides a simplified way to configure a LinuxONE server. It supports Linux and KVM systems with FCP-attached SCSI storage. It removes the need for IBM Z configuration tools, such as Hardware Configuration Definition (HCD) and Hardware Configuration Manager (HCM).

The entire system configuration is performed from the HMC. It eliminates the need to build a stand-alone IOCP input deck for I/O devices. It also eliminates the need for the second power-on reset (POR) to enable dynamic I/O, which is required in non-DPM configurations. DPM provides for fully dynamic reconfiguration of CPU, memory, and I/O resources. Adding and dynamically reconfiguring I/O devices is greatly simplified with DPM.

Simplified vision of DPM

DPM provides the following advantages:

- Fast. Much faster than managing with HCD and HCM. From hours to minutes.
- Easy. Intuitive user interface. No need for multiple administrators with different skills or tools. Do not expect First In Enterprise Linux customers to adopt the previous way.
- Powerful. The same efficient PR/SM hardware virtualization without the complexity. It supports dynamic configuration changes with a few mouse clicks. It also provides a foundation for “bare metal” cloud.

> **Note:** For more information about DPM operations and capabilities, see *z Systems IBM Dynamic Partition Manager (DPM) Guide*, SB10-7168.

2.2 Choosing a z/VM installation method

As you proceed through this section, it is suggested that you have a copy of *z/VM installation guide*, GC24-6292 for reference.

In the z/VM 7.2.0 Installation Guide, the following methodologies are provided to install z/VM:

- Traditional first level:
  - VMSSI installation to DASD
  - Non-VMSSI (stand-alone) installation to:
    - DASD
    - FCP/SCSI LUNs
- Traditional second level non-VMSSI (stand-alone) installation to:
  - DASD
  - FCP/SCSI LUNs
- Upgrade VMSSI installation to DASD
- Upgrade non-VMSSI (stand-alone) installation to:
  - DASD
  - FCP/SCSI LUNs

Overall, the z/VM 7.2.0 Installation Guide classifies all installations under two main categories, *Traditional* or *Upgrade*.

2.2.1 Understanding traditional and upgrade installations

In this section, we describe traditional installations of z/VM versus upgrade installations.

Traditional (new) installations

In the context of installing z/VM, the word *traditional* is confusing to many who are new to z/VM. To help clarify, whenever you see mention of *traditional installation*, imagine instead that the text says *new installation*. The traditional method installs a new z/VM system (stand-alone or as the first member of a VMSSI cluster), which can then be customized according to your needs

Upgrade installations

An upgrade installation is used to perform an upgrade-in-place installation to migrate from z/VM 6.4 or 7.1 to z/VM 7.2.

A new release system is used as a staging system that is installed in a second level system of your current system. This upgrade-in-place process essentially performs the following tasks:

- Creates a clone of your current system to run as a second level system.
- Upgrades the cloned system to z/VM Version 7 Release 2.
- After upgrade is complete, moves the new level of code to the current system with minimal impact.

If your current system is a VMSSI cluster, you can upgrade one member individually.

2.2.2 Classifications used in this book

Although the two installation methods that are described in the preview section are effective, for further clarity, the authors of this book instead choose to group installations under the following main categories:

- New installation to:
  - DASD
  - FCP/SCSI
- Upgrade installation to:
  - DASD
  - FCP/SCSI

If you are planning to install z/VM in either of the following ways, use the *z/VM 7.2 Installation Guide*, GC24-6292, instead of this Virtualization Cookbook because we do not address these options:

- *Second level* (z/VM virtualized as a guest under z/VM)
- On Fibre Channel Protocol (FCP)/SCSI disk

2.2.3 New and upgrade installations to DASD

In this book, we describe the installation process by covering the initial load of z/VM in memory first. Regardless of which installation method you choose, the initial load in memory is the same process.

From that point forward, you decide which type of the following installations is better for you:

- A first-level VMSSI installation of z/VM from DVD or FTP server onto 3390 DASD (highly recommended at least with one VMSSI member only)
- A first-level Non-SSI installation of z/VM from DVD or FTP server onto 3390 DASD

If you are new to z/VM, plan to install on only one logical partition (LPAR), and plan to use ECKD DASD. It is still recommended that you proceed by using the VMSSI path. The installation of a single-member VMSSI cluster provides you with an easy path for future expansion to add member nodes later.

If you are installing to SCSI disk, you cannot install with VMSSI enabled. However, non-SSI installations can use the *z/VM Centralized Service Management* (VMCSM) capability that was introduced with z/VM 7.2. VMCSM allows non-SSI installations to manage service across a set of systems from one central system.

In this book, we do not describe the non-VMSSI installation in detail; however, we describe setting up VMCSM in Chapter 9, “z/VM Centralized Service Management” on page 275.

2.2.4 Installing as VMSSI with live guest relocation

As previously described in 1.4.2, “Software” on page 20, z/VM VMSSI with live guest relocation (LGR) makes it possible to actively run Linux virtual servers to be relocated from one z/VM system to another within the same VMSSI cluster.

Before you adopt VMSSI, you must understand the architectural, disk, memory, and networking requirements.

Even if you are experienced with the installation and service of z/VM, it is important that you read the instructions for installation of z/VM with or without VMSSI.

If you are experienced with z/VM, or, are migrating from an older release, be aware that *significant* changes were made throughout the platform, beginning with Version 6 Release 2. It is critically important that you understand what changed and why.

Consider the following points:

- A VMSSI cluster must have direct logical links between all systems.

  All VMSSI clusters use Inter-System Facility for Communications (ISFC) for intra-cluster communication for LGR. ISFC uses channel-to-channel (CTC) devices. For maximum throughput, when you are setting up your network, follow the section, “Guidelines for planning your network in an SSI cluster”, in Chapter 2 of *z/VM Getting Started with Linux on System z*, SC24-6194. Faster CTC speeds increase throughput and result in shorter relocations.

- As a preferred practice, define the same real device numbers to reference the same devices on all members of the VMSSI cluster.

  Discuss this practice with your hardware administrator to ensure that this naming is reflected by the I/O configuration data set (IOCDS).

Attention: The z/VM Single System Image feature enables sharing the configuration, parameters, and directory data over Channel-to-channel adapter (CTCA) connections, When you choose to install a system as second-level during the z/VM installation process, customized modifications that were made to the generated system configuration and user directory parameters exist.

Because of these differences, you must not create a cluster that contains a mix of first-level and second-level z/VM members. Attempting to do so results in unpredictable or catastrophic results.

For more information, see *z/VM CP Planning and Administration, Version 7 Release 2*, SC24-6178.

Factors that can affect relocation, system performance, or both

Consider the following factors in planning for Linux LGR:

- VM memory: The size and use of the VM's memory can affect relocation performance. Parts of the processing for relocation are proportional to the size of the VM. The cost of this processing increases with larger VMs. Relocation performance also is affected by the frequency and amount of memory that is being changed in the VM.
- Matching VM configurations: To prepare for LGR, ensure that the VM has a configuration that allows for it to be relocated and that a matching configuration can be set up on the destination member. For information about configuration requirements and about verifying a VM's eligibility to relocate, see Chapter 27 of *IBM z/VM CP Planning and Administration,* SC24-6178.
- CPU utilization: The z/VM V7.2 SSI feature synchronizes all of the members in the cluster. You must ensure that you allocated enough system resources to account for the necessary synchronization and communication among members. After initialization, the synchronization overhead is relatively low. Communication between members increases during negotiations for access to devices and other resources, and during LGR. For example, two independent systems that run fine at peak utilization (close to 100%) might experience performance problems when they are joined in a cluster.

  For z/VM members that are running as a second-level z/VM system, they must not be waiting for CPU more than 10% of the time. For more information, see the "Resource Limit Conditions" section of Chapter 27 of *IBM z/VM CP Planning and Administration,* SC24-6178.
- Paging and other system resources: To prepare for LGR, the target system must have enough system resources during and after the relocation. You must ensure that your paging space is adequate. z/VM 7.2 changed the capabilities and effects of the **`CP SET RESERVED`** command and you must consider new information during your planning.
- To be safe, you need twice as much available space as the total virtual memory that can be defined on the system. The easiest way to check this aspect of system resources is to issue the **`CP QUERY ALLOC PAGE`** command, which shows the percent that is used, the slots that are available, and the slots that are in use.

  If you add the size of the VMs that are being relocated (a 4 KB page = a 4 KB slot) to the slots in use, and that brings the in-use percentage over 50%, that is likely a situation where you consider adding paging volumes. This query command provides only a snapshot in time. A utility that is called `vir2real` is available from the IBM VM Download Library that is helpful in making a determination about whether sufficient paging is in place. For more information about the IBM VM Download Library, see 15.1, "Installing a package from the IBM VM Download Library" on page 440.

- Real memory: Real memory resources are important for the source and the destination systems for relocations. You need enough real memory to hold buffers during the relocation on both systems, and accommodate the incoming guest's working set afterward on the target system. Relocation performance also is affected by the level of overall resource constraint for the source and destination systems.
- Dump space: If you are allocating a large amount of real storage (memory), you must plan for dump space so that if you must collect a system dump, enough space is available to write it to disk.
- Linux distributions and LGR: With the introduction of LGR among members of your VMSSI cluster, it is increasingly important to identify the level of Linux on IBM Z that is running within each member. The latest level of a distribution release is considered to be the supported level by the Linux Distribution Partners. The preferred practice for setting up VMSSI is to ensure that you are running on the latest level and that your distribution is supported by your Linux distributor.

Why ECKD DASD is required

If z/VM 7.2 is installed into a VMSSI, at least one extended count key data (ECKD) volume is necessary for the Persistent Data Record (PDR).

If you plan to implement RACF, the database must be configured as being shared and at least two ECKD DASD volumes are necessary. Concurrent virtual and real reserve/release must always be used for the RACF database DASD when RACF is installed in A VMSSI.

For more information about sharing a RACF database, see *z/VM RACF Security Server System Programmer's Guide*, SC24-6212.

For information about DASD sharing, see *IBM z/VM CP Planning and Administration,* SC24-6178.

2.2.5 Planning aids

Regardless of which method you plan to use, always check this z/VM web page for current information, and make sure all requirements are satisfied. Review the following topics at this web page:

- Important z/VM Installation News
- z/VM Installation Tips
- Preventive Service Planning (PSP) bucket for z/VM 720 installation.

If you are not familiar with the Hardware Management Console (HMC) and z/VM, refer to the official *z/VM 7.2 Installation Guide* as you progress through the installation process.

We encourage you to download and use the following IBM publications:

- Chapter 25 of *IBM z/VM CP Planning and Administration*, Version 7 Release 2, SC24-6178
- *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- *z/VM: Getting Started with Linux on IBM Z*, Version 7 Release 2, SC24-6287

2.3 Bill of materials

The resources that are needed for a Linux on IBM Z or LinuxONE system project are grouped into the following types:

- Hardware
- Software
- Networking

2.3.1 Hardware

The following hardware is needed:

- A minimum of one, and up to a maximum of four LPARs with:
  - Processors or CPUs per LPAR: One IFL (or CP) minimum; two or more are recommended.
  - Memory: First level installation requires at least 768 MB of real storage that is assigned to the LPAR where z/VM is to be installed. A total of 8 GB storage or more is recommended.
  - DASD: A total of 20 3390-09s were allocated to our lab system that is described in this book.
  - Open Systems Adapter (OSA) network cards: One card minimum with six device numbers. Two OSA Express cards with six device numbers are recommended for high availability.
- A network-attached computer running Linux or UNIX that acts as a File Transfer Protocol (FTP) server with at least 8 GB of disk space.

> **Important:** FTP servers that run on any DOS or DOS-like operating system, such as Microsoft Windows, are not supported. Therefore, Linux or UNIX (such as IBM AIX®) is recommended. Be aware that a Windows-based FTP server is likely to cause code page translation, which results in corruption issues during network transport.

- A workstation that includes network access to the IBM Z and operates as the HMC if physical access is not possible.

2.3.2 Software

z/VM 7.2 installation media with documentation is needed. The physical media of DVDs are described. If you use *Shopz* to download the z/VM installation media and make it available by using an FTP server, physical media is not needed.

After installing z/VM, you need one or more of the following software resources to install Linux:

- Red Hat Enterprise Linux version 7.1 or greater (we used version 8.2) installation media. If you do not have it, you can request a 180-day evaluation copy from Red Hat. For more information, see Appendix , “Online resources” on page 524.
- SUSE Linux Enterprise Server version 15 or later installation media (DVD .iso files). If you do not have it, you can request a 180-day evaluation copy at no charge from SUSE. For more information, see Appendix , “Online resources” on page 524.

- Ubuntu Server version 20.04 LTS or later installation media. If you do not have it, you can obtain a full copy at no charge from Canonical. For more information, see Appendix , “Online resources” on page 524.
- The code that is associated with this book, as described in Appendix C, “Additional material” on page 489
- Tools on the workstation:
  - A 3270 emulator, such as x3270, Attachmate Extra, Hummingbird Host Explorer, or IBM Personal Communications.
  - A Linux Secure Shell (SSH) customer, such as PuTTY, xTerm, or similar.
  - A Virtual Network Computing (VNC) viewer, such as RealVNC, TightVNC, MobaXTerm, or similar.

2.3.3 Networking

The following network resources are needed:

- One TCP/IP address for each:
  - z/VM system
  - Linux virtual server
- Associated TCP/IP information:
  - Primary Domain Name Server (DNS) IP address and fully-qualified host name.
  - Secondary DNS IP address and fully-qualified host name.
  - DNS sub-domain/domain names to use for z/VM and Linux virtual servers (these might not always be the same).
  - DNS server TCP/IP address.
  - TCP/IP gateway.
  - TCP/IP subnet mask.
  - TCP/IP maximum transmission unit (MTU) size.

    Be sure to review the information that is provided in this book regarding correct MTU sizes on IBM z/Architecture Open Systems Adapters (OSAs)

  The TCP/IP addresses must be routed to the appropriate OSA cards.
- Virtual LAN (VLAN) ID if you plan to use a VLAN
- *Locally Administered* Ethernet Media Access Control (MAC) address range to be used across all members of the z/VM cluster (IEEE 802 OSI layer 2 EUI-48; `02:00:00` OUI).

  Under most circumstances, this information is obtained from the person or team that is responsible for managing your network. If no one is responsible for managing your network, use the following guidelines for assigning a Locally Administered address range to use with your SSI cluster:
  - The address range must be unique. Ideally, it must be unique across your entire enterprise. If not possible, at a minimum it must be unique within the LAN segment to which your OSA cards are cabled.

    If you are deploying your new z/VM infrastructure onto a network segment that shares a Server Access or Server Distribution switch with any existing production systems, it is important that you ensure the uniqueness of your MAC address range. If it is not unique, severe negative consequences can occur to the other network-attached devices or systems.

  - Each MAC address consists of a 12-digit hexadecimal number, which *must* begin with `02` as the first octet; for example: "*020C46005501*".
  - Do not assign "0000 0000 0000" or "FFFF FFFF FFFF."
  - The range is 0200 0000 0000 - 02FF FFFF FFFF.

The MAC address range must be unique within the LAN segment to which you connect. For more information, see this IEEE web page.

> **IMPORTANT:** If you plan to install Ubuntu Server by using Volume 4 of this book series, review Appendix C of *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354 for considerations regarding systems that are behind a network proxy or firewall.

2.4 Disk planning

For the scope of this book, a *storage administrator* is defined as a person or group, such as a department, team, or even a third-party vendor, that is specifically tasked with one or more enterprise disk/flash storage functions, such as architecture, operations, configuration, management, and business controls.

> **Important:** If you have a storage administrator that meets our definition, you must involve them in your planning activities. Never assume anything; therefore, if you are not sure, check with your CIO office to determine if such a person or group exists.

Your storage administrator is crucial to the success of your z/VM cloud deployment. By involving them early on in the planning process, you ensure a smooth deployment that performs at its optimal best.

2.4.1 Primary considerations

Consider different aspects when you plan how to choose and allocate disk storage, including the following factors:

- Conventional ECKD DASD versus fixed-block architecture (FBA) disks over Small Computer System Interface-Fibre Channel Protocol (SCSI-FCP)
- Model (size) of type 3390 ECKD DASD disks:
  - Standard models are:
    - **3390-03**: 3390 Model 3, or "Mod-3s"

      3390-03 are no longer supported for z/VM installation, and are too small for any practical Linux usage. Because of these factors, this book does not use or discuss this model.
    - **3390-09**: 3390 Model 9, or "Mod-9s",
    - **3390-27**: 3390 Model 27, or "Mod-27s",
    - **3390-54**: 3390 Model 54, or "Mod-54s",
  - Oversize models, or volumes larger than 3390-54: 3390-A Extended Address Volumes (EAVs) that are larger than Mod-54s.
- Amount of disk storage per Linux image and how to allocate file systems.

DASD versus SCSI-FCP

This book describes how to use conventional ECKD DASD and to access SCSI-FCP and emulated DASD (EDEV) disks.

SCSI/FCP disks require worldwide port name/logical unit number (WWPN/LUN) identifiers and the correct zoning setup.

Frequently, a combination of these types of disk storage is used. When a combination is used, the ECKD or EDEV DASD are often used for the root file system and SCSI/FCP disks are used for large data storage areas.

3390-09s, or larger

Emulated 3390-09s format to approximately 6.8 GB, about three times larger are 3390-27s (20.1 GB), and 3390-54s format to about 42.1 GB. z/VM 7.2 can be installed on to 3390-09s or larger. However, for the z/VM *system volumes* that are used for version, residence, release, and so forth, we recommend the use of 3390-09.

For PAGE and SPOOL areas, DASDs larger than 3390-09 can be used; however, all of the involved page and spool disks *must* be of identical size and not serve mixed purposes. If you have 3390-54 or larger volumes available that also include associated HyperPAV aliases, we recommend starting out with these for PAGE and SPOOL use.

Specific larger IBM Z or LinuxONE system customer environments are choosing to use volumes that are larger than 3390-27s to avoid reaching the 64 K limit of real device addresses (four-character hexadecimal).

EDEVICEs

SCSI Disks are emulated as 9336 Model 20 FBA DASD. FBA Emulation allows any operating system or application that supports a 9336 to use SCSI disks without change. The following Emulated 9336 disk sizes are supported:

- 1 TB for CP with the exception that PAGE, SPOL, and DRCT allocations must remain below the 64 GB (minus a 4 K page) mark on a CP-formatted volume because internal addressing of these slots is limited to 224 4-K pages.
- 381 GB for CMS/GCS including software functions that depend on CMS functions, such as DIRMAINT MOVE, COPY, ERASE and DFSMS MOVE, COPY, and CHECK.

z/VM officially supports the following IBM hardware as emulated 9336 DASD (other SCSI disks might work because a generic SCSI driver is provided):

- IBM DS8000
- IBM DS6000
- ESS 750/800
- SAN Volume Controller
- IBM XIV®

Consider the following points when the use of EDEVICEs is considered:

- A path length increase exists with EDEVICE I/O processing compared to CKD. In addition, overhead exists that is associated with the translation that must occur, which results in CPU consumption. However, depending on your workload, it is possible that either one of these factors can be a negligible or trivial amount.
- It is recommended to use dedicated FCP subchannels for Linux virtual servers unless a requirement exists for any one or more of the benefits of sharing, deployment, and maintenance simplification, and minidisk caching that come with z/VM minidisk usage.

- Use traditional ECKD DASD for paging and spooling, except when little paging or spooling activity exists, or sufficient processor resources are available to handle increased path length.
- Continue to use traditional ECKD DASD for CMS minidisks, except when minidisk cache hit ratios are high, or sufficient processor resources are available to handle increased path length.
- Consider EDEVICEs when no ECKD DASD is available, minidisks are required, or large disks are necessary.

For more information about the studies that led to these recommendations, see the following resources:

- CP Disk I/O Performance
- Linux Disk I/O Alternatives
- SCSI Performance

Disk storage for each Linux image

This version of the Virtualization Cookbook now recommends two 3390-09 DASDs that are attached as "`01 to END`" full-pack minidisks at virtual address `0700` and `0701`, which gives about 13.5 GB of space.

You might choose to add another full pack 3390-9 DASD at address `0701`, which doubles the disk space to 13.6 GB.

We strongly recommend the layout of file systems for Linux virtual servers that is shown in Figure 2-2 and listed in Table 2-1 on page 46.

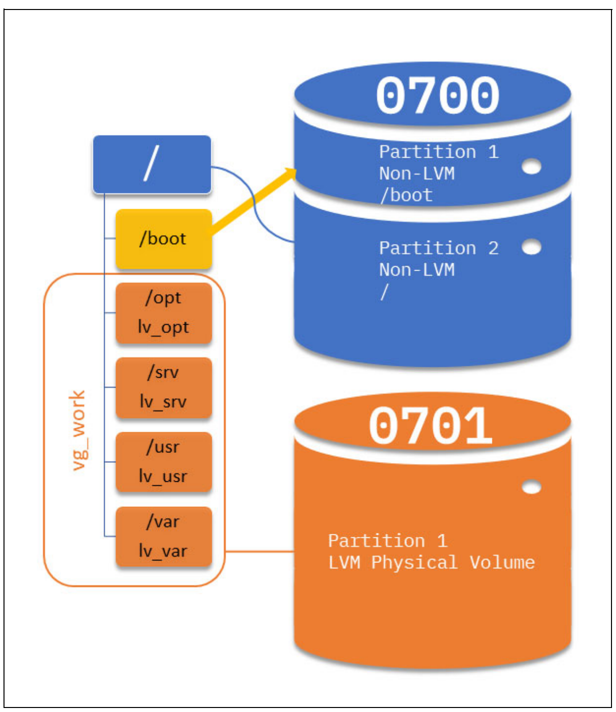


Figure 2-2 File system layout

Table 2-1 Linux virtual server file system layout

File system	Size	Mount point	Disk partition / logical volume	File system type	LVM
boot	`512 MiB`	`/boot`	`ccw-0.0.0700-part1`	ext4 or xfs	No
system root	`approx 6.3 GiB`	`/`	`ccw-0.0.0700-part2`	Use the recommended type for your distribution.	No
opt	`1.5GiB`	`/opt`	`vg_work_lv_opt`	xfs or ext4	Yes
srv	`512MiB`	`/srv`	`vg_work_lv_srv`	xfs or ext4	Yes
usr	`2.5GiB`	`/usr`	`vg_work_lv_usr`	xfs or ext4	Yes
var	`1.0GiB`	`/var`	`vg_work_lv_var`	xfs	Yes

We recommend this layout for the following reasons:

- The intention of this book is to teach suitable methods that are based on long-established good practices. Therefore, we are choosing to correctly segment the file systems in a way that provides stability, easier recovery from potential mistakes or accidents, and affords easy expansion when necessary
- One of the single most prevalent bad practices is to deploy a Linux system with everything in one file system. Frequently, we hear LVM used as a justification for having done so. Consider the following points:
  - A false belief exists that deploying in one file system that uses Logical Volume Manager (LVM) mitigates the future trap this creates. This belief is not true.
  - When the root (/) file system becomes full, Linux kernel panics and crashes. An LVM saves you only from this situation if good monitoring is used and enough time is available to react. In most cases, administrators do not have enough time to react before a crash occurs.
- The size of your root file system, even as workload expands because good practice dictates that applications and data go into one of the four file systems we created as LVM logical volumes in Table 2-1.
- As application and system data becomes more voluminous, you can quickly and easily expand `vg_work` by adding another minidisk as a physical volume. This process can be done in minutes and does not require an outage.
- Recovery of a Linux system that uses LVM for the root file system is a needless complexity that can be avoided. The process to mount, access, and make repairs to Linux that is rooted in an LVM is a \ manual procedure that is ripe with the possibilities for human error to make things worse than they might already be.
- Linux distributions that use BTRFS or ZFS for the root (/) file system often include features that are enabled that are not suitable for application and work data.

File system types

We recommend the use of the default file system type for your Linux distribution for the root file system (/):

- RHEL 8: xfs
- SLES 15: Btrfs
- Ubuntu 20.04: xfs

Important: Never use BTRFS for application or work data. Although the subvolume and snapshotting features that are included with BTRFS are excellent for recovery of accidental changes or mistakes to the root file system, they are not designed for use outside of the core components of a distribution present after a clean, newly-installed system. Using BTRFS for `/home`, `/opt`, `/srv`, `/usr`, or `/var` quickly becomes a severe performance bottleneck that often results in a system failure to adequately process workload.

For more information about the types of file systems that are available for each Linux distribution, see the following resources:

- Red Hat: Managing file systems
- SUSE Release Notes: 5.6.1 Comparison of Supported File Systems
- Ubuntu: Linux Filesystems Explained

Swapping

The terms *swap disk* and *swapping* are often used in describing the operation of Linux, but it is important to realize that Linux does not “swap” process spaces in the traditional sense. The Linux behavior that is commonly referred to as *swapping* is the paging of data out of system memory to a block device based on least recently used (LRU) algorithms. For historical reasons, the term *swap* is still shown and used, but it refers to paging in Linux. Swap is also used to help delineate paging at the Linux level (swap disk), in contrast to paging at the z/VM level.

z/VM includes a feature that is called *virtual disks* (VDisks). VDisks exist in memory but are presented to guest operating systems as disks. IBM Z memory is fast (many times faster than disk especially); therefore, the use of VDisks for swap spaces makes sense because they are so fast.

Ideally, your Linux systems never have to swap, but workloads cannot be predicted so easily. Therefore, all Linux VMs must have an adequate set of swap spaces. What defines “an adequate set of swap space” is debatable. However, the z/VM and Linux community agrees that one or two small swap spaces on a VDisk that also can be backed up by a larger swap space on real disk are best.

Real-world experience from IBM ATS and Lab Services teams that work with customers indicates that correctly sized Linux guests seldom, if ever, swaps. This book describes how to set up two VDisk swap spaces, but not the additional physical disk because it is seldom required.

To create the swap spaces, the `SWAPGEN EXEC` created by Sine Nomine Associates, Inc. is used. It creates and formats a Linux swap disk from Conversational Monitor System (CMS). For more information about the current version, and instructions for downloading and installing `SWAPGEN` are in the `swapgen-readme.txt` file that is available on this Sine Nomine web page.

2.5 HiperDispatch planning

In z/VM 6.3, IBM introduced new virtual server dispatching technology into z/VM this is called *HiperDispatch*. The z/VM HiperDispatch enhancement is meant to help the workload realize good performance from the CPC's memory subsystem, especially from its caches. To achieve this performance, z/VM HiperDispatch runs the partition in vertical mode, dispatches virtual servers in a topologically aware way, and uses logical CPUs in accordance with the availability of physical CPU power.

Workloads that are likely to benefit from z/VM HiperDispatch are those workloads for which cache performance is likely to influence total performance. Workloads with these traits usually involve a few CPU-heavy virtual servers for which isolating their execution from one another in the physical hardware will allow cache to adapt well to the respective servers' memory reference habits.

For more information about HiperDispatch, see this web page.

2.6 Storage planning

As you proceed through this section, remember the terminology that was covered in section 1.4, "A high-level overview of components and terminology" on page 18. *Central storage* and *real storage* both refer to main system memory.

Memory planning might be the most complex issue with z/VM and Linux on IBM Z or LinuxONE systems, yet it is the most important factor in performance.

By using the Hardware Management Console (HMC), you can define the `INITIAL` and `RESERVED` real storage (memory) for each LPAR that will run z/VM. When you IPL z/VM, the control program assumes that all of the `INITIAL` central storage is available to it.

2.6.1 z/VM 7.2 initial installation and migrations considerations

Consider the following important points about z/VM 7.2:

- Previous versions of z/VM required that you allocate a lesser amount of `INITIAL` real storage to the LPAR for installation (typically 8 GB or less). With z/VM 7.2, this requirement no longer exists.
- If you are migrating to z/VM 7.2 from an older version, the LPAR activation profile might be configured to define Expanded Storage, or `XSTORE`. As of z/VM 6.2, IBM no longer recommends XSTORE as an auxiliary paging device, and usage of XSTORE with z/VM 6.4 was at that point considered to be deprecated.

  The aging and filtering functions that were provided by XSTORE are now provided by z/VM's Global Aging List (GAL).

  If your LPAR has XSTORE that is defined for it, convert your XSTORE to real storage and then, run the system with no XSTORE at all. For example, if you ran an earlier version of z/VM in a 32 GB partition with 4 GB of XSTORE, you change to 36 GB of real storage with no XSTORE when migrating to z/VM 7.2.

- z/VM 7.2 changed the capabilities and effects of the **CP SET RESERVED** command; therefore, review the systems that you are migrating to ensure the values that are in use are still suitable. Earlier editions of z/VM sometimes failed to honor **CP SET RESERVED** settings for VMs, which prompted users to oversize the amount of reserved storage that they specified. z/VM 7.2 is more effective and precise in honoring reserved settings.
- z/VM 7.2 also permits **CP SET RESERVED** for Named Saved Systems (NSS) or Discontiguous Saved Segments (DCSS). This new capability was especially intended for the MONDCSS segment.

  In previous z/VM releases, MONDCSS was at risk of being paged out and consequently unavailable for catching control program (CP) Monitor records under heavy storage constraint. Because CP Monitor records are especially needed when the system is under duress, IBM suggests that you establish a reserved setting for MONDCSS. Use a reserved setting that is equal to the size of MONDCSS to ensure residency for the instantiated pages of MONDCSS.

2.6.2 Storage allocation

If you have no previous IBM Z experience, you might think that the simplest solution is to over-allocate `INITIAL` storage with the expectation that an overabundance results in z/VM never paging and Linux never swapping. It is likely that not only are such resources often not available, but also that over-provisioning results in needless rework or potential performance problems for you in the future.

For more information about memory planning, see the following resources:

- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- IBM z/VM Performance Resources
- For z/VM 6.3 and newer Releases

After you determine the `INITIAL` value that you plan to use for each LPAR, also always define `RESERVED` storage, even if you think that you will not require it immediately. By defining `RESERVED` storage, you can dynamically increase the amount of memory that is available to z/VM without requiring a shutdown. The added storage is typically obtained from *standby storage*, which is a dynamically calculated value that is based on the following information:

- Amount of storage that is installed in the CPC that is not claimed by activated LPARs
- Amount of `RESERVED` storage that is specified for the LPAR

A good rule is to allocate memory on a "just enough" basis for each Linux VM. A good starting point is to set a VM size by changing the memory allocation value to be slightly above the value at which Linux has more than 64 MB of combined cache and buffer. If you are migrating workloads from distributed platforms, you typically find that Linux under z/VM needs less memory to which you are accustomed.

In addition to the "just enough" amount of memory, assign a number of VDisks as SWAP disks to each of the Linux VMs. The VDisks must provide as much memory as Linux needs at the peak level during operation.

When you are performing calculations for z/VM page disks, add up all of the maximum real storage and VDisks, plus reserve. That amount also must be available as real memory to the LPAR, plus page space.

One recommended rule is to have as few VMs logged on (or in a disconnected state) as possible to handle the workload that is presented. Every VM that is not required needs to be logged off where suitable because more memory is available for the other VMs that remain running.

2.6.3 Global aging list

In the unusual situation where your CPC has an abundance of unassigned memory and you can assign enough initial storage to your LPAR to fit your intended z/VM workload entirely into central storage, run with a small global aging list and with global aging list early writes disabled.

In all other situations, use the IBM recommendation to run the system with the default global aging list size. For the environment that is described in this book, the default is used.

The global aging list can be controlled by using the **CP SET AGELIST** command or the **STORAGE AGELIST** system configuration file statement.

2.7 Paging

Your paging channels and DASD must be planned to be equipped for conducting multiple concurrent paging I/O operations. As the paging configuration becomes capable of increasing levels of I/O concurrency, CP then becomes increasingly able to handle the concurrent execution of page-fault-inducing VMs. This ability results in optimal throughput for your workloads.

2.7.1 Recommendations, tips, and hints

Consider the following recommendations, tips, and hints when you create paging channels and DASD allocations:

- No mixing: A disk volume must be all paging (cylinders 1 to END) or no paging at all. Never allocate paging space on a volume that also holds any other data, such as spool space, user minidisks, or anything else.
- Match up: Make all of your volumes the same size. If you decided to use 3390-09s, all your paging volumes must be 3390-09. If you decide to use all 3390-54s (as we did for this book), all of the paging volumes must then be 3390-54s. This rule applies to whatever type of volumes you ultimately choose.

  When the volumes are unequally sized, any smaller volumes fill up and become ineligible as targets for future page-outs. This situation results in an unnecessary bottleneck that negatively affects system performance by restricting the z/VM opportunity for paging I/O concurrency.
- Use HyperPAV paging or spread out if not: If you are planning to deploy workloads that are frequently paged, ensure your paging strategy is up to par.

  If your paging volumes are HyperPAV capable, ensure they have sufficient HPAV Aliases associated. For a 3390-54, a minimum of eight aliases is recommended.

  If you are not using HyperPAV-capable DASD, it is imperative that you instead spread your paging space over as many smaller volumes as possible. Get many little paging volumes instead of one or two large paging volumes. The more paging volumes that you provide, the more paging I/Os z/VM can run concurrently. The method that is used in this book of allocating multiple 3390-9s is consistent with this recommendation.

- Select the best subsystem: If multiple disk storage subsystems in your data center are accessible to z/VM, carefully consider which of these disk storage subsystems you select for the z/VM paging volumes. Disk storage subsystem controllers of different speeds, cache sizes, capabilities, and existing loads are all considerations when you decide where to place paging volumes.
- Performance matters: Within a certain disk storage subsystem controller, volume performance is sensitive to how the volumes are placed. Avoid poor volume placement, such as putting all of your paging volumes into one rank or other similar situations, by involving your SAN or disk management team in the planning process.
- Increase your channel-path identifiers (CHPIDs): If you can, it is highly recommended that you run multiple CHPIDs to each DASD controller that holds paging volumes. Consider two, four, or eight CHPIDs per controller, even if IBM Fibre Channel connection (FICON) is used because this approach can substantially increase throughput.
- If Fibre Channel Protocol (FCP) CHPIDs exist and SCSI DASD controllers are installed, you might consider them for paging. A SCSI logical unit number (LUN) that is defined to the z/VM system as an EDEV and is attached by using ATTACH to SYSTEM for paging allows the z/VM control program to overlap I/Os to it. You can achieve paging I/O concurrency without the need for multiple volumes. However, this approach includes a penalty of increased processor cycles. If you are CPU-constrained, do not take this route. The number of physical unit (PU) cycles that are required for each I/O to perform EDEV versus traditional ECKD is not a trivial number.
- Avoid Enterprise Systems Connection (ESCON) CHPIDs: An ESCON CHPID can carry only one I/O at a time. FICON CHPIDs can run multiple I/Os concurrently (32 or 64), depending on the generation of the FICON card.
- Reserve a few slots in the SYSTEM CONFIG CP-owned list, so that you can add paging volumes without an IPL, if necessary.

2.7.2 Calculating paging space

The z/VM 7.2 edition of *IBM z/VM CP Planning and Administration,* SC24-6178, was updated and now includes a new formula for calculating the amount of paging space to allocate. The formula is shown in Table 2-2.

Table 2-2 Calculations for paging planning

A	Sum of primary address space sizes for all logged-on VMs, which are all VMs that are typically expected to be up and running under normal conditions, such as DirMaint and other service machines, and the Linux VMs.
B	Sum of the sizes of any data spaces for all logged-on VMs.
C	Sum of the sizes of any VDISKs that are created for all logged-on VMs.
D	Sum of the sizes of any shared NSSes or DCSSes for all logged-on VMs.
E subtotal	Add A + B + C + D to obtain a subtotal of E.
F	Multiply E by `1.01` to account for the DAT structures that are associated with all that pageable data.
G	Total number of CP directory pages that are reported by DIRECTXA. Be sure that you convert pages to MB, GB, or whatever unit that is used.

A	Sum of primary address space sizes for all logged-on VMs, which are all VMs that are typically expected to be up and running under normal conditions, such as DirMaint and other service machines, and the Linux VMs.
H	10% of real storage that is defined for the LPAR to allow for system-owned virtual pages.
I **minimum**	Add E + G + H to obtain I, which is your bare minimum paging space amount. When you calculate this value, you determine the bare minimum paging space amount that is ordinarily considered safe.
J **recommended**	Multiply I by a reserve margin value to obtain J, which is the final value you use. Reserve is added because your calculation might be uncertain and your system grows over time. Multiply your value that is calculated for I by a reserve safety margin to help to protect yourself against abends that are caused by paging space filling up. IBM offers no rule for the reserve safety factor multiplier that you need to use. For this book, the authors chose a value of 25% and recommend no less than 15%.

2.8 Passwords and passphrases

Good passwords are critical to good security. However, requiring many different passwords and high levels of complexity generally leads to users writing them down, which detracts from good security. Sometimes, it is difficult to balance these two extremes.

This book considers different system programming and administration roles:

- z/VM systems programmer

  Sometimes also referred to as a *sysprog,* they can also perform the functions of an administrator and more.

- Linux systems administrator
- Linux VM users

The z/VM systems programmer and Linux systems administrator can be the same person.

The method of backing up z/VM data onto the Linux administration system that is described in this book means that the Linux administrator can access all z/VM passwords, unless RACF/VM or another external security manager (ESM) is in place to provide password encryption. As described in 1.5, “Choices and decisions for this book” on page 27, the use of an ESM must not be seen as optional.

Important: A different method of backing up z/VM data, such as IBM Backup & Restore Manager for z/VM, must be chosen in a situation where the following conditions exist:

- An ESM is not used.
- The z/VM systems programmer and Linux systems administrator roles are separate.
- The Linux administrator does not have access to the clear text z/VM password.

If you choose to deploy your system in an unsecured manner by omitting an ESM, the `USER DIRECT` file and possibly others that are associated with it \ contain clear text passwords for every USER and IDENTITY.

Previous versions of this book set all z/VM and Linux system administration passwords to the same trivial password. Such a practice is no longer acceptable, even in a test or other non-production environment.

In general, it is considered bad practice to set a password for any USER or IDENTITY that is a non-person ID. That is, each individual who requires access to z/VM must be issued their own ID that is for their exclusive use and never to be shared with anyone else for any reason.

Each individual's ID should then be granted LOGONBY authority for any other ID to which they require access. This process provides an audit trail to identify who accessed an alternative ID and when. It also eliminates the need to change and remember multiple passwords.

The following non-person system administration IDs must be set to LBYONLY:

- z/VM systems programming IDs: `MAINT`, `MAINT720`, `PMAINT`,
- The z/VM network administrator: `TCPMAINT`
- The Linux VM users (with or without access to 3270 sessions, and with or without the root passwords)

For more information about how to correctly configure LBYONLY, see 6.13.6, "Creating a time-based virtual service machine named CRONSVM" on page 212.

2.9 Network

Most servers need network connectivity. On IBM Z or LinuxONE systems, different networking methods are available, which are provided by the I/O subsystem that controls the LPARs, such as IBM PR/SM (possibly with IBM Dynamic Partition Manager [DPM]), or the z/VM hypervisor.

Several other networking methods are available that are not described in this book because they are impractical, complex, or limited.

The virtual Open Systems Adapter (OSA) and Virtual Switch (VSWITCH) TCP/IP devices that are used in z/VM are equivalent to high-end enterprise class networking devices. You need at least a fundamental understanding of TCP/IP networking, routing, and switching to plan your network configuration more easily.

If you have a new installation, we recommend that you use one of the options that are described in this section.

2.9.1 Involvement of stakeholders

For the scope of this book, we are defining *network administrator* as a person or group, such as a department, team, or even a third-party vendor, that is specifically tasked with one or more network functions, such as architecture, operations, configuration, management, and business controls.

> **Important:** If you have a network administrator that meets our definition, you must involve them in your network planning tasks. Never assume anything; if you are unsure whether a network administrator person or groups exists, check with your CIO office.

Even the best system is useless if nobody can use it. Without networking, your systems cannot be accessed. As such, it is important that you view your network administrator as an ongoing partner in all of your IT projects.

Collaboration ensures that MAC addresses and other network configuration do not become a problem that can cause more re-configuration efforts later. This collaboration also help to ensure that you do not inadvertently create any possible problems.

2.9.2 Open Systems Adapters

For use with z/VM and Linux in the ways described in this book, Open Systems Adapters (OSAs) are needed, which are a type of OSD. An OSA in OSD mode is operating in queued direct input/output (QDIO).

IBM developed the QDIO mode to enhance high-speed OSA Express adapters. It brings with it many features, such as fast-path I/O, direct memory addressing, LPAR-to-LPAR capabilities, and configuration-from-host options.

The following rules apply to configuring OSA OSD adapters for use:

- READ/WRITE must be an even/odd pair:
  - READ is set to an even device number.
  - WRITE is set to the device number after READ.
- DATA can be another device number on the same device, it is nearly always the device number following the WRITE device.

A good method to provide this type of device is the read, write, data method:

- The first device gets `0AD0`, `0AD1`, and `0AD2`.
- The second device gets `0AD4`, `0AD5`, `0AD6` or `0AD6`, `0AD7`, and `0AD8`.

2.9.3 Network attachment options and considerations

The use of z/VM Virtual Network Switches (VSWITCH) is the IBM recommended practice. It features less computational expense than the other alternatives. When it is correctly configured, it provides built-in fail-over. It also supports 802.1q VLAN by port, 802.1q VLAN by user, port isolation, and 802.3ad link aggregation.

VSWITCH interfaces

The VSWITCH type of network is provided to z/VM VM systems by z/VM. During the installation phase, a basic VSWITCH is configured. VSWITCHES are software switches that offer many of the capabilities that are provided by a real switch.

The IBM recommended configuration for z/VM VSWITCHES is ETHERNET mode, which is also frequently referred to as *Layer 2*. Ethernet-based networks are currently used for most z/VM installations that run Linux as a VM operating system. Under virtually all circumstances, z/VM VSWITCHES must be configured as ETHERNET/Layer 2.

Characteristics of VSWITCH interfaces

VSWITCH interfaces feature the following characteristics:

- Are run by a set of redundant virtual service machines, by default.
- Can fail over with up to three real devices.
- Can be configured to be virtual LAN (VLAN)-aware.
- Up to 2,048 virtual network interfaces can be coupled to a single VSWITCH.

- Ports on VSWITCHES can be configured as USER-based or with port numbers.
- Access and trunk ports can be configured for VSWITCHES.
- VSWITCH network interfaces always operate on port 0 of the virtual device.

Direct-attached Open Systems Adapter

Linux uses the identical drivers to run a direct-attached OSA that it uses to run VSWITCH interfaces. With a direct-attached OSA, you get the fastest network connection to the external network.

> **Important:** The direct-attached OSA configuration is not a recommended configuration. Use it with extreme caution. It is a single point of failure in an otherwise highly available environment. It can also become packed and stall, triggering rollbacks or failures in fault-intolerant consumers. The VSWITCH does not exhibit this weakness and remains the recommended method.

Direct-attached OSA network devices characteristics

The administrator must be aware of the following issues about the use of a direct-attached OSA:

- If a dual-port OSA is attached directly to a VM, the VM can choose whether it wants to configure either of the physical ports.
- All OSA ports can be reused if they are shared over different LPARs.
- The OSA ports within an LPAR can be used only once, which also holds true for z/VM configurations, such as Port Groups.
- Linux does not use PORTNAMES. They can be omitted or set as an empty string.
- If the separation between several ports of an OSA is important, do not use direct-attached OSA devices.

2.9.4 Maximum transmission unit size matters

In this section, we describe sizing the maximum transmission unit (MTU). An MTU is the size of the largest single unit of information transmitted along a network. Sizing the MTU correctly can improve bulk protocol throughput.

MTU sizes for QDIO Open Systems Adapters

MTU sizing is different from traditional distributed systems. Set MTUs to the maximum size that is supported by all hops on the path to the final destination to avoid fragmentation.

A simplified way to help determine the correct MTU size to use is to run **`tracepath destination`** from a Linux system on the same network segment that your OSAs use. It traces the path to destination and discovers MTU along this path by using User Datagram Protocol (UDP). It is similar to **`traceroute`**, but it does not require superuser (root) privileges. Follow these guidelines:

- If the application data is less than or equal to 1400 bytes, use an MTU size of `1492`.
- If the application is able to send larger chunks of data, use an MTU size of `8992`.

> **Important:** These MTU sizes are all-inclusive. They apply to z/VM and Linux virtual servers running under z/VM.

Transmission Control Protocol (TCP) uses the MTU for the window size calculation, not the actual application send size. For VSWITCH, an MTU size of 8992 is recommended if possible because an OSA is optimized for use with an 8992 MTU. With synchronous operation, SIGA is required for every packet. You do not encounter packing (stalling due to queuing) from a VSWITCH as you do from a dedicated OSA.

Attention: Most Linux distributions assuming that they are operating on commodity hardware; therefore, the default Ethernet MTU of 1500 is the typical default value. Do not use the default MTU of 1500 with Linux on IBM Z or LinuxONE; instead, use the guidance that is provided in this chapter.

For more information about TCP/IP and the z/VM hypervisor, see Appendix B, “Reference, cheat sheets, blank worksheets, and education” on page 475.

For more information about Open Systems Adapters, see *IBM z Systems Connectivity Handbook*, SG24-5444.

2.9.5 IBM HiperSockets

IBM HiperSockets (HIPERS) are a networking option that is defined in the I/O configuration of the LPAR. This type of network connection provides an internal memory-to-memory connection within one Channel Subsystem (CSS).

The main use for this type of network option is as an internal connection between two Z LPARs that are on the same physical system frame. An example is the connectivity between a z/OS LPAR that serves IBM DB2® and a z/VM LPAR where one or more Linux VMs access the DB2 instance on z/OS.

With the advent of 10 GB Ethernet connectivity, the utility from HIPERS is diminished. The network speeds that are obtained from a HiperSocket are substantially similar to that of 10 GB Ethernet.

It is the general opinion of the authors of this book, whenever you can use 10 GB Ethernet or HiperSockets, use the 10 GB Ethernet unless a substantially compelling reason exists not to use it.

HiperSockets characteristics

HiperSockets feature the following characteristics:

- They are used to configure the connection between two Z LPARs.
- HIPERS is a direct memory-to-memory pipeline; it operates at memory transfer speed.
- HIPERS can use System Assist Processors (SAP).
- HiperSockets require access to the General Purpose Command Processor (CP). Because of this, HiperSockets are not available on any LinuxONE CPC, nor any LPAR that is defined to have only IFL engines.
- HIPERS can become queued and stall if it is waiting for processor time.
- Bridges to networks that are outside of the mainframe can be configured.
- The interconnection between Linux and z/OS is configured with Layer 3 HiperSockets.
- HIPERS provides a limited number of real devices that cannot be reused from different LPARs.

- As with an OSA connection, a triplet of devices must be dedicated.
- Unlike an OSA connection, off-loading is not possible.

> **Important:** Use caution when you implement HIPERS on processor-bound, heavily loaded LPARs where software or programs are running that are sensitive to or intolerant of network delays or stalls.

2.9.6 IPv4 and IPv6

If you do not use IPv6, your Linux virtual servers should be deployed so that IPv6 is not initialized for the network adapters you use. Many users are unaware that modern Linux systems always attempt to use the IPv6 TCP/IP stack first, and then fall back to IPv4 if unsuccessful. Each time this attempt occurs, a penalty is paid in extra processor time and in some cases, latency that might not be considered trivial in all circumstances.

2.10 Channel-to-channel adapter planning

It is important to plan adequate channel-to-channel (CTC) definitions to achieve an adequate LGR quiesce and relocation time. At an absolute minimum, it is recommended that two CTC devices are connected for each SSI member through two channel paths. During the SSI installation process, you can install only two CTCs for each SSI member.

If you \ use the VMSSI feature for consolidated systems management only, two CTCs for each SSI member are sufficient. If you plan to use LGR also, plan to add a third CTC to each member node soon as the workload grows. For more information, see 5.8, "Adding CTCAs to an SSI cluster" on page 130.

To configure the two channel-to-channel adapters (CTCAs) during initial installation, you need Input/Output Definition File (IODF or IODEF) information from your hardware configuration colleague. They initially must provide two FICON Native CTC (FCTC) control units, each with a minimum of four devices.

Example 2-1 shows sample IODF configuration statements that represent the FCTC connections for member 1.

Example 2-1 Sample IODF configuration statements for member 1

```
CNTLUNIT CUNUMBR=47E0,PATH=((CSS(0),4C)),UNITADD=((00,004)),  *
      LINK=((CSS(0),0E)),CUADD=2E,UNIT=FCTC
IODEVICE ADDRESS=(47E0,004),UNITADD=00,CUNUMBR=(47E0),        *
      STADET=Y,PARTITION=((CSS(0),A02)),UNIT=FCTC
CNTLUNIT CUNUMBR=57E0,PATH=((CSS(0),4D)),UNITADD=((00,004)),  *
      LINK=((CSS(0),0A)),CUADD=2E,UNIT=FCTC
IODEVICE ADDRESS=(57E0,004),UNITADD=00,CUNUMBR=(57E0),        *
      STADET=Y,PARTITION=((CSS(0),A02)),UNIT=FCTC
```

Example 2-2 shows the configuration for member 2.

Example 2-2 Sample IODF configuration statements for member 2

```
CNTLUNIT CUNUMBR=4120,PATH=((CSS(2),4C)),UNITADD=((00,004)),   *
      LINK=((CSS(2),31)),CUADD=2,UNIT=FCTC
IODEVICE ADDRESS=(4120,004),UNITADD=00,CUNUMBR=(4120),         *
      STADET=Y,PARTITION=((CSS(2),A2E)),UNIT=FCTC
CNTLUNIT CUNUMBR=5120,PATH=((CSS(2),4D)),UNITADD=((00,004)),   *
      LINK=((CSS(2),30)),CUADD=2,UNIT=FCTC
IODEVICE ADDRESS=(5120,004),UNITADD=00,CUNUMBR=(5120),         *
      STADET=Y,PARTITION=((CSS(2),A2E)),UNIT=FCTC
```

From the provided CTC information, the selected CTC devices from `ITSOZVM1` are `41A0` and `41A1`. For `ITSOZVM2` devices, `5190` and `5191` are used.

For more information about CTC capacity recommendations, see *z/VM Performance Update for z/VM 6.2*.

2.11 z/VM standardized naming conventions

It is in your best interest to adopt and use standardized conventions wherever possible so that you and others can recognize z/VM resources by their names. This section describes several standardized conventions.

2.11.1 DASD volume labeling convention

You must adopt a standardized convention for labeling DASD. If one or more IBM Z or LinuxONE systems exist, your IT department might use a labeling standard that determines the labels to be given to the DASD that is used by your z/VM LPARs.

Each DASD includes a real device address that consists of four hexadecimal digits, and each DASD has a six character label. When assigning DASD labels, include the four-digit address in the label so that you can easily tell the address of each DASD from its label. As your workload expands and you begin to work on disk administration tasks, this information proves to be invaluable because it saves time.

When followed thoroughly, this naming convention ensures that no two DASDs have the same label, which can be important, especially when an IBM z/OS LPAR can access the DASD.

Sometimes, DASD is shared among LPARs. In this case, your z/VM LPAR can see DASD that is owned by other LPARs. In this situation, it is convenient to identify the LPAR or SSI that owns the DASD.

Therefore, the volume labeling convention that is used in this book creates DASD labels where each of the six characters specifically represents identifying information:

Character 1 — Identifies the LPAR or VMSSI cluster name or ID. The example VMSSI cluster in this book is identified by the character “V”.

Character 2 — The basic function of the DASD.

Characters 3-6 — The real device number of the DASD volume.

> **Attention:** The authors of this book strongly encourage configuring the IODEF in a manner so that each LPAR can see only the DASD that it owns or to which it requires access. This idea can be thought of as installing a tall fence around a parcel of land you own so that a clear boundary exists. Configuring the IODEF in this manner helps eliminate risk. Configure your IODEF in a way that all LPARs can see.

First character in the label

The letter "V" is hardcoded into the **CPFORMAT** REXX EXEC in the tarball file that is associated with this book, which is described in Appendix C, "Additional material" on page 489. This EXEC uses this volume labeling convention. If you want to use a different LPAR identifier character, you can easily change them (search for the `firstChar` variable). The following line is the pertinent line of code:

```
/***************************************************************/
...
  Address COMMAND
  firstchar = 'V'
...
```

Second character in the label

The following characters are used for the types of DASD in the second character of the label:

M	Minidisk space (PERM)
P	Paging space (PAGE)
R	RACF database volume
S	Spool space (SPOL)
T	Temporary disk space (TDISK)
V	z/VM operating system volumes

2.11.2 Virtual network device naming convention

> **Note:** Previous iterations of this book recommended the use of `0600` as the default virtual device number for a virtual network adapter. We recommend against this default because it can become confusing.

Default virtual network adapters should use `0AD0`, which can be thought of as "adapter 0". By reserving the `0AD0` - `0ADF` range for networking only, it is easy to recognize a virtual NIC.

Each virtual NIC uses three channels; read control, write control, and data. Therefore, when you define a virtual NIC that uses device `0AD0`, it also implies that `0AD1` and `0AD2` also are used. Issuing the **QUERY VIRTUAL NIC *device* DETAILS** command displays the output that is shown in Example 2-3.

Example 2-3 Query virtual NIC with details

```
CP QUERY VIRTUAL NIC OADO DETAILS
Adapter 0AD0.P00 Type: QDIO      Name: HYD1G1      Devices: 3
  MAC: 02-04-0D-00-00-58         VSWITCH: SYSTEM VSWITCH3
      PQUplinkTX: Normal
      RX Packets: 892254     Discarded: 0          Errors: 0
      TX Packets: 134108     Discarded: 0          Errors: 2
      RX Bytes: 1303867677           TX Bytes: 8170432
  Connection Name: HALLOLE   State: Session Established
```

```
→ Device: 0AD0  Unit: 000   Role: CTL-READ
→ Device: 0AD1  Unit: 001   Role: CTL-WRITE
→ Device: 0AD2  Unit: 002   Role: DATA       Port: 2205
     Options: Broadcast Multicast IPv6 IPv4 VLAN
       Unicast IP Addresses:
         9.60.87.203          MAC: 02-04-0D-00-00-58
         FE80::204:D00:200:58 MAC: 02-04-0D-00-00-58 Local
       Multicast IP Addresses:
         224.0.0.1            MAC: 01-00-5E-00-00-01
         FF01::1              MAC: 33-33-00-00-00-01 Local
         FF02::1              MAC: 33-33-00-00-00-01 Local
         FF02::1:FF00:58      MAC: 33-33-FF-00-00-58 Local
```

Because 0AD0 - 0AD2 are used for your first NIC, if you must define a secondary NIC by using 0AD3.

> **Note:** The previous edition of this book used 0600 as the default virtual device for Virtual NICs. We recommend any user that is still using this default to discontinue it and instead adopt the advice that is presented in this section.

2.11.3 Minidisk and virtual disk naming convention for Linux

> **Note:** Previous iterations of this book recommended the use of 0100 as the first virtual device number for a minidisk to be used with a Linux virtual server. We recommend against the use of the range 0100 - 06FF because it competes with z/VM volumes in the same range.

To ensure it is clear that a minidisk or virtual disk is used for Linux, begin at virtual device 0700. This device makes it possible to easily understand what is a Linux volume versus something else.

Avoid the use of the range 0100 - **06FF** for Linux minidisks or dedicated volumes.

2.11.4 Backup file naming convention

It is recommend that you keep copies of important original z/VM and Linux configuration files in case you need to reference them. Because z/VM file names are limited to 16 characters (eight for the file name and eight for the file type), only the last four characters of the file type are used, which often requires characters to be overwritten.

Originals

For the original file, the suffix ORIG is used. For example, before any editing is done, the original USER DIRECT file is copied to the file USER DIREORIG before it is modified for the first time. The original SYSTEM CONFIG file is copied to the SYSTEM CONFORIG file. This process ensures that you always retain the original copy.

Recent versions

The following commonly adopted practices are used for retaining previous versions of files:

- The “it works” method.
- The “now minus” method.

Both methods are described next and use the `SYSTEM CONFIG` file as an example to help you decide which makes the most sense for you:

- It works:
  - Only the most recent working copy is retained and uses the suffix `WRKS` (for “it WORKS”).
  - Before editing, `SYSTEM CONFIG` is copied to the file `SYSTEM CONFWRKS`.

  In this fashion, a copy of the original, a copy of the current, and the last working copy of configuration files always exist.
- Now minus; typically referred to as *n minus*:
  - Several recent working copies are retained and use the suffixes `-1`, `-2`, `-3`, and so on.
  - Before editing, `SYSTEM CONFIG` is copied to the file `SYSTEM CONF-1`.
  - If `SYSTEM CONF-1` exists, it is renamed to the file `SYSTEM CONF-2 first`.

  This method is simple to implement by using REXX.

2.11.5 Command retrieve convention

The ability to retrieve past commands is a common tool. Often, it is helpful to retrieve in both directions if you overlook the command for which you are looking. The default Linux shell, bash, performs this task by default by using the up arrow and down arrow keys.

A convention in z/VM is to use the F12 function key (labeled PF12 on physical 3270 devices) to retrieve the last command, although it is not defined to all VMs. No convention exists to retrieve commands in the other direction, but it is possible to set another key to that function. Therefore, the F11 key is used to retrieve forward because it is next to the F12 key. Also, the same function is useful in the editor, **`XEDIT`**. The **`?`** subcommand retrieves past commands,; therefore, it is recommended that you assign it to F12.

For more information about implementing this concept, see “PROFILE EXEC for Linux virtual machines” on page 497.

2.12 Architectural overview of this book's environment

Figure 2-3 shows a block diagram with the CPC, LPARs, and volume labels that are used in this book. The example VMSSI in this book consists of two members on a single CPC; therefore, the bottom half of the diagram is left blank.

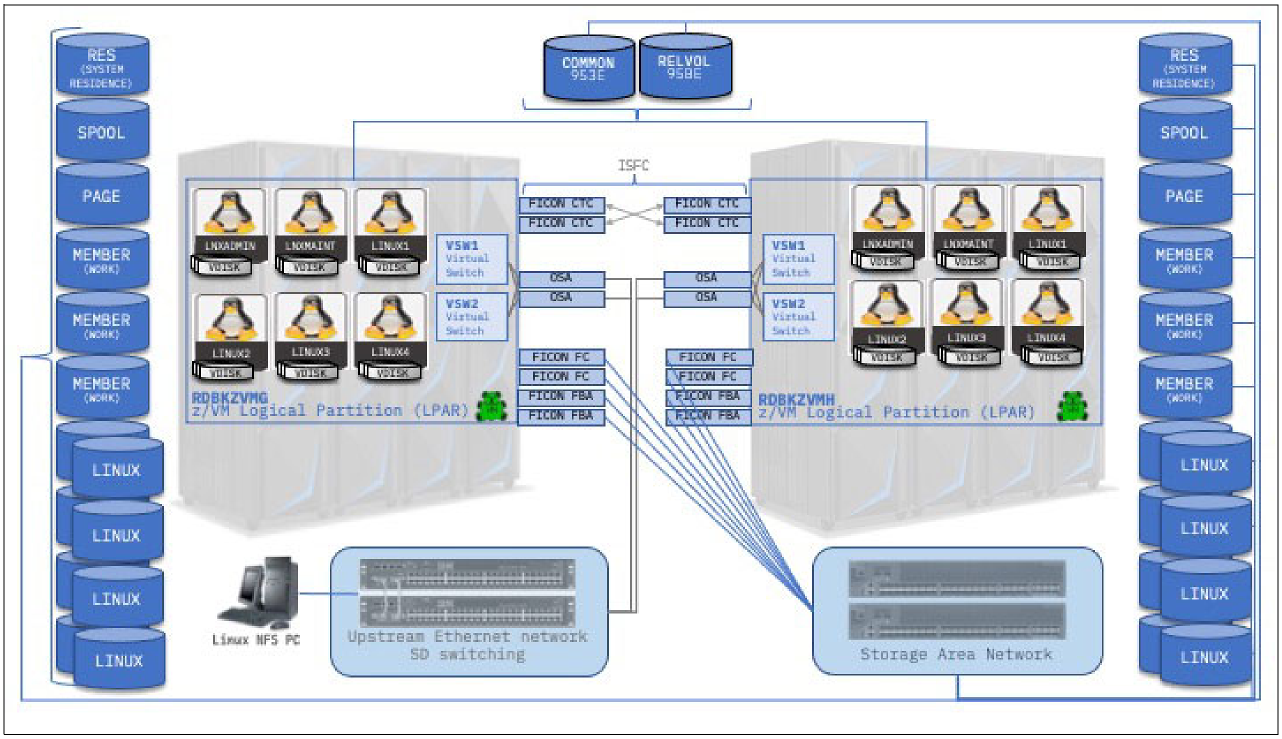


Figure 2-3 IBM Z environment that is used in this book

2.13 Example planning worksheet

The following tables make up the overall planning worksheet that is used to install and configure a z/VM 7.2 SSI cluster, Linux guests, and any required supporting resources.

The planning worksheet that is shown includes the resources that were used in writing this book to serve as an example.

A corresponding blank planning worksheet is provided for your use in Appendix B, “Reference, cheat sheets, blank worksheets, and education” on page 475.

> **Important:** The values in the following fully populated worksheet tables are shown in ***`monospace bold italics`*** to signify that you need to replace the ***`example value`*** with the correct value for your environment.

Previous versions of this book encouraged the collection of passwords on the planning worksheets. For obvious reasons, this suggestion is not followed. No valid reason exists to write down passwords. Instead, catalog all of the user IDs and passwords you are responsible for in a well-known, thoroughly vetted trust no one (TNO) password vault solution, such as LastPass, OnePass, or other similar password management solution.

2.13.1 IBM Shop Z

If you are ordering z/VM by using Shop Z, as described in 5.1, "Obtaining z/VM through electronic download" on page 98, use Table 2-3 to record the values that you use.

Table 2-3 Shop Z data

Name	Value
User ID	*My IBMid*
Password	*MyPassword (obtained from my password manager / vault)*
Order number	*U06074293000*
Order name	*Products - z/VM Version 720 - 2020-09-21*
Date/Time	*2020-09-21 13:15 EDT*

2.13.2 Hardware Management Console

In section 5.4, "Starting the z/VM installation" on page 111, you see how to start a z/VM installation from the Hardware Management Console (HMC). Complete Table B-3 on page 483 to document the values that you use.

The values that are required for each row are listed in Table 2-4.

Table 2-4 HMC data

Name	Value
HMC location or URL	*https://end3hmc6.atslab.endicott.ibm.com*
HMC user ID	*pwnovak*
HMC password	Because out HMC is set up to use LDAP by using an RACF-LDAP connector, this password is the same password that is used to log in to z/VM
FTP source system	*atsendftp.atslab.endicott.ibm.com*
z/VM installation directory	*/srv/zvm/720*

2.13.3 z/VM installation planning panels

You must document the information for your environment before you use an installation planning panel (INSTPLAN).

INSTPLAN panels 1 and 2

As described in section 5.5.1, "Copying the in-memory z/VM system to DASD" on page 115, the **INSTPLAN** command is run from the Integrated 3270 Console. \

The required values for each row are listed in Table 2-5.

Table 2-5 INSTPLAN values for the first two panels

Name	Value	Comment
Product install target	☐ **F** (SFS file pool) ☐ **M** (minidisk)	Leave set to the default value of **F** for all
Language	☐ AMENG ☐ USENG	Select AMENG (American English)
DASD model	☐ 3390 **10016** ☐ 3390 ________	3390 10016 is a reference to 3390 Model-9. If you are installing to a larger disk size, overtype 10016 with the cylinder count for the disks you will use. (Installation to a fixed-block architecture (FBA) disk is not described in this book.).
File pool name	**VMPSFS**	VMPSFS (default) recommended.
System type	☐ SSI (VMSSI) ☐ Non-SSI	
Non-SSI system name		Used for non-SSI installation only.
Number of members		VMSSI installation only (usually 2 or 4).
SSI cluster name	***RBVMCLA***	VMSSI installation only. Our example is Redbooks z/VM cluster "A"
Automatic configuration	**NO**	

INSTPLAN panel 3

Complete this table to document the values that you use on the third installation panel, as described in 5.6.1, "Copying in-memory z/VM system to DASD" on page 124. The member names become the z/VM system identifiers, and the LPAR names must be the same names as on the HMC.

The required values for each row are listed in Table 2-6.

Table 2-6 INSTPLAN values for panel 3

Slot	Member name	LPAR name	Comment
1	***RDBKZVMG***	***A09***	Member 1 system identifier and LPAR name
2	***RDBKZVMH***	***A0A***	Member 2 system identifier and LPAR name
3	***RDBKZVMI***	***A0C***	Member 3 system identifier and LPAR name
4	***RDBKZVMJ***	***A0D***	Member 4 system identifier and LPAR name

INSTPLAN worksheet 3

Complete the worksheet in Table B-6 on page 484 to document the volume labels and real device addresses that you use on the third installation panel that is described in 5.5.1, "Copying the in-memory z/VM system to DASD" on page 115.

The values that are required for each row are listed in Table 2-7.

Table 2-7 INSTPLAN values for volume definitions

Type/purpose	Label	Address	Comment
COMMON	***AV953E***	***953E***	Common volume
RELVOL	***AV95BE***	***95BE***	Release volume
SESVOL	***AVFAAE***	***FAAE***	VMSES/E service volume
SFSVOL	***AVFFA0***	***FFA0***	LNX SFS pool volume
Mem 1 RES	***AV963E***	***963E***	Member 1 residence volume
Mem 1 SPOOL	***AS96BE***	***96BE***	Member 1 spool volume
Mem 1 PAGE	***AP973E***	***973E***	Member 1 page volume
Mem 1 WORK	***AMFF10***	***FF10***	Member 1 work volume
Mem 2 RES	***AV97BE***	***97BE***	Member 2 residence volume
Mem 2 SPOOL	***AS983E***	***983E***	Member 2 spool volume
Mem 2 PAGE	***AP98BE***	***98BE***	Member 2 page volume
Mem 2 WORK	***AMFF20***	***FF20***	Member 2 work volume
Mem 3 RES	***AV9F00***	***9F00***	Member 3 residence volume
Mem 3 SPOOL	***AS9F01***	***9F01***	Member 3 spool volume
Mem 3 PAGE	***AP9F02***	***9F02***	Member 3 page volume
Mem 3 WORK	***AM9F02***	***9F02***	`Member 3 work volume`
Mem 4 RES	***AV9F03***	***9F03***	Member 4 residence volume
Mem 4 SPOOL	***AS9F04***	***9F04***	Member 4 spool volume
Mem 4 PAGE	***AP9F05***	***9F05***	Member 4 page volume
Mem 4 WORK	***AM9F06***	***9F06***	`Member 4 work volume`

INSTPLAN worksheet 4

The values in Table 2-8 document the common volume and CTC addresses that are used in this book. This pane is shown in 5.5.1, “Copying the in-memory z/VM system to DASD” on page 115. If only two members exist in the SSI, you must specify only two pairs of CTCAs (from member 1 to member 2, and vice versa).

Table 2-8 INSTPLAN values for channel-to-channel adapter definitions

CTC device addresses:			
From member 1		**From member 2**	
To: member 1	N/A	To: member 1	***2790*** ***4790***
To: member 2	***4D80*** ***2D80***	To: member 2	N/A

2.13.4 z/VM networking resources

Table 2-9 lists the networking resources that are used in the examples in this book. They are needed when you start the **IPWIZARD** and when you create a VSWITCH for the Linux VMs.

Table 2-9 z/VM and networking resources

Name	Value	Comment
TCP/IP user ID	TCPIP	TCPIP is recommended.
z/VM host name, member 1	***RDBKZVMG***	
z/VM host name, member 2	***RDBKZVMH***	
TCP/IP domain name	***cpolab.ibm.com***	System domain name is usually set in DNS.
TCP/IP gateway	***9.76.61.1***	The router to and from the local subnet.
DNS server 1	***9.0.130.50***	Obtain from network administrator.
DNS server 2	***9.0.128.50***	Obtain from network administrator.
DNS server 3	***9.60.70.80***	Obtain from network administrator.
Interface name	***OSAETH0***	OSAETH0 is recommended. The first character is the letter O. The last character is the numeral 0 (zero).
OSA starting device number	***1944***	Start of OSA *triplet* for z/VM TCP/IP stack.
Subnet mask Subnet CIDR mask	***255.255.255.0*** ***/24***	Assigned by your network administrator.
OSA device type	QDIO (layer2)	
VLAN ID		Obtain from network administrator if required.
MTU size	8992	1492 or 8992 for jumbo frames. Recommend 8992 with PMTUD on.
VSWITCH1 primary OSA triplet	***0FA0*** ***0FA1 0FA2***	Specify the first real device number and the next two device numbers will also be used.
VSWITCH1 second OSA triplet	***19A0*** ***19A1 19A2***	Ideally, it needs to be on a different CHPID.
VMLAN MAC prefix, member 1	***0200A1***	Assigned by your network administrator.
VMLAN MAC prefix, member 2	***0200A2***	Assigned by your network administrator.

Important: For setting the VMLAN MACPREFIX value, *IBM z/VM CP Planning and Administration,* SC24-6178, includes the following information:

> "In A VMSSI cluster, system-defined locally administered MAC addresses are created by using the prefix value that is specified on the MACPREFIX operand. The MACPREFIX value must be different for each member of the cluster. The default value is 02*xxxx*, where *xxxx* is the member's slot number on the SSI statement. If the MACPREFIX value is explicitly defined, the VMLAN statement must be qualified for the member to which it applies. Therefore, if a VMLAN statement with the MACPREFIX operand is retained from the non-SSI system or created in this step, it must be qualified for member VMSYS01."

2.13.5 z/VM DASD

Table 2-10 lists the z/VM DASD resource values that are used in the examples in this book.

Table 2-10 z/VM DASD that is used in this book

Device	Label	Type	Notes
FFA1	*VMFFA1*	System (3390-27)	LNX: SFS, MONWRITE, LXW minidisks
FF11	*VMFF11*	System (3390-09)	LNXADMIN 0700 on member 1 (RHEL)
FF12	*VMFF12*	System (3390-27)	LNXADMIN 0701 on member 1 (RHEL)
FF21	*VMFF21*	System (3390-09)	LNXADMIN 0700 on member 2 (SLES)
FF22	*VMFF22*	System (3390-27)	LNXADMIN 0701 on member 2 (SLES)
FF31	*VMFF31*	System (3390-09)	LNXADMIN 0700 on member 3 (Ubuntu)
FF32	*VMFF32*	System (3390-27)	LNXADMIN 0701 on member 3 (Ubuntu)
FF50	*VMFF50*	System (3390-09)	MINIDISKS - POOL1
FF51	*VMFF51*	System (3390-09)	MINIDISKS - POOL1
FF52	*VMFF52*	System (3390-09)	MINIDISKS - POOL1
FF53	*VMFF53*	System (3390-09)	MINIDISKS - POOL1
E300		(EDEV 10 GB)	LNXADMIN 0750 on member 1 (RHEL)
E301		(EDEV 10 GB)	LNXADMIN 0750 on member 2 (SLES)
E302		(EDEV 10 GB)	LNXADMIN 0750 on member 3 (Ubuntu)
E303		(EDEV 10 GB)	LINUX1 0750 (RHEL)
E304		(EDEV 10 GB)	LINUX2 0750 (RHEL)
E305		(EDEV 10 GB)	LINUX3 0750 (SLES)
E306		(EDEV 10 GB)	LINUX4 0750 (SLES)
E307		(EDEV 10 GB)	LINUX5 0750 (Ubuntu)
E308		(EDEV 10 GB)	LINUX6 0750 (Ubuntu)
E309		(EDEV 10 GB)	

2.13.6 FCP devices

Table 2-11 and Table 2-12 list the z/VM FCP resource values that are used in the examples in this book.

Table 2-11 EDEV LUN assignments

Device	LPAR	WWPN	Storage WWPN	LUN/RDEV
B800	A09	C05076DD90000480	500507630500C74C 50050763050BC74C	4010401100000000/3000 4010401200000000/3001 4010401300000000/3002 4010401500000000/3003 4010401600000000/3004 4011401100000000/3005 4011401200000000/3006 4011401300000000/3007 4011401400000000/3008 4011401500000000/3009
	A0A	C05076DD90000400		
B900	A09	C05076DD90000A60	500507630510C74C 50050763051BC74C	
	A0A	C05076DD90000AE0		

Table 2-12 Direct-attached FCP assignments

N_Port ID Virtualization (NPIV) device	LPAR	NPIV WWPN	LUN	Used by
B801 B901	A09	C05076DD90000404 C05076DD90000A64	4010401700000000	LINUX1
	A0A	C05076DD90000484 C05076DD90000AE4		
B802 B902	A09	C05076DD90000408 C05076DD90000A68	4010401800000000	LINUX2
	A0A	C05076DD90000488 C05076DD90000AE8		
B803 B903	A09	C05076DD9000040C C05076DD90000A6C	4010401900000000	LINUX3
	A0A	C05076DD9000048C C05076DD90000AEC		
B804 B904	A09	C05076DD90000410 C05076DD90000A70	4010401A00000000	LINUX4
	A0A	C05076DD90000490 C05076DD90000AF0		
B805 B905	A09	C05076DD90000414 C05076DD90000A74	4010401B00000000	LINUX5
	A0A	C05076DD90000494 C05076DD90000AF4		
B806 B906	A09	C05076DD90000418 C05076DD90000A78	4011401600000000	LINUX6
	A0A	C05076DD90000498 C05076DD90000AF8		

N_Port ID Virtualization (NPIV) device	LPAR	NPIV WWPN	LUN	Used by
B807 B907	A09	C05076DD9000041C C05076DD90000A7C	4011401700000000	
	A0A	C05076DD9000049C C05076DD90000AFC		
B808 B908	A09	C05076DD90000420 C05076DD90000A80	4011401800000000	LNXADMIN on Member 1 (RHEL)
B809 B909	A0A	C05076DD900004A4 C05076DD90000B04	4011401900000000	LNXADMIN on Member 2 (SLES)
B80A B90A	A0C	C05076DD900004F6 C05076DD90000C06	4011401A00000000	LNXADMIN on Member 3 (Ubuntu)

2.13.7 Linux resources

Table 2-13 and Table 2-14 list the Linux resources that were used in this book.

Table 2-13 External Linux FTP server resources that were used in this book

Name	Example Value	Comment
TCP/IP address	***9.60.86.4***	
User/password	***ftpuser/linux4vm***	
FTP installation source directory	**/srv/linux/rhel/8** **/srv/linux/sles/15** **/srv/linux/ubuntu/20.04**	Directory with DVD 1 of each distribution

Table 2-14 Linux common configuration values that were used in this book

Name	Example Value	Comment
Linux root password	***xxxxxxx***	Do not use a trivial password.
TCP/IP gateway	***9.60.86.1***	Obtain from the network administrator.
Subnet mask	***255.255.254.0 or /23***	Obtain from the network administrator.
DNS servers	***9.60.70.80, 9.60.70.81***	Obtain from the network administrator.
Virtual Network Computing (VNC) installation password	***xxxxxxx***	Must be eight characters.

2.13.8 Host names and IP addresses

Table 2-15 lists the host names and associated IP addresses that are used in the examples in this book.

Table 2-15 Hosts that are used in this book

Host name	IP address	VM	Notes
rdbkzvmg.cpolab.ibm.com	***9.76.61.243***	LPAR A09	z/VM 7.2 SSI member 1
rdbkzvmh.cpolab.ibm.com	***9.76.61.244***	LPAR A0A	z/VM 7.2 SSI member 2
rdbkzvmi.cpolab.ibm.com	***9.76.61.245***	LPAR A0C	z/VM 7.2 SSI member 3
vmlnx2-1.ats.ibm.com	***9.60.101.90***	LNXADMIN	RHEL FTP Server
vmlnx2-2.ats.ibm.com	***9.60.101.91***	LNXADMIN	SLES FTP Server
vmlnx2-3.ats.ibm.com	***9.60.101.92***	LNXADMIN	Ubuntu FTP Server
vmlnx2-4.ats.ibm.com	***9.60.101.93***		
vmlnx2-5.ats.ibm.com	***9.60.101.94***	LINUX1	RHEL
vmlnx2-6.ats.ibm.com	***9.60.101.95***	LINUX2	RHEL
vmlnx2-7.ats.ibm.com	***9.60.101.96***	LINUX3	SLES
vmlnx2-8.ats.ibm.com	***9.60.101.97***	LINUX4	SLES
vmlnx2-9.ats.ibm.com	***9.60.101.98***	LINUX5	Ubuntu
vmlnx2-10.ats.ibm.com	***9.60.101.99***	LINUX6	Ubuntu
vmlnx2-11.ats.ibm.com	***9.60.101.100***		
vmlnx2-12.ats.ibm.com	***9.60.101.101***		
vmlnx2-13.ats.ibm.com	***9.60.101.102***		
vmlnx2-14.ats.ibm.com	***9.60.101.103***		
vmlnx2-15.ats.ibm.com	***9.60.101.104***		

Chapter 3

Security considerations

Modern computing systems must be operated in ways that provide protection from various security risks. z/VM provides several controls and facilities that can be used to enhance the security protection of the Linux systems it hosts.

In this chapter, we discuss security considerations for your z/VM environment that help to make it a more secure place for Linux systems to run.

This chapter includes the following topics:

3.1 Security policy

Every organization must have a policy for managing enterprise-wide security. This policy covers all aspects of the organization's security: from physical building security and access through to information security.

Establishing rules up-front, getting support for those rules at the corporate level, and knowing where the rules are written down, makes it much easier to work out what you need so that those rules are enforced in your IT systems.

This book does not discuss the establishment of a corporate security policy. However, we do cover some aspects of z/VM security that are relevant to supporting good IT security policy and protecting Linux virtual machines (VMs) under z/VM.

3.2 External Security Manager

An External Security Manager (ESM) is a user-supplied program with which you define your system's own security mechanism for preventing unauthorized user access to resources from application programs, and the unauthorized initiation of transactions.

Linux features its own security model, which is implemented in several kernel and file system capabilities, including the following examples:

- File system permissions, and access control lists (ACLs)
- Mandatory Access Control layers, such as SELinux or AppArmor
- Cgroups

Because Linux features its own security capability, it is tempting to consider security at the hypervisor layer to be superfluous or unnecessary. However, this idea is a significant and risky security exposure that can put your environment at risk.

3.2.1 How hypervisor security protects you

To illustrate the exposure of inadequate security at the hypervisor layer, we consider an environment where several virtualized systems run under the same hypervisor instance. If the hypervisor does not adequately protect the resources of one VM from another, data can be moved between VMs that are outside the control of the VM operating system.

This movement of data between VMs might occur through attaching the disks of one instance to another, or by creating an isolated network connection between them. These methods might result in sensitive data (such as SPI) being exfiltrated from a high-sensitivity VM to one with a lower security profile.

One way to mitigate this issue might be to create separate hypervisor instances for the different security levels in your environment. However, this process quickly leads to problems in scalability and management by forcing more hypervisor instances that might be needed to support the workload. Also, it becomes harder to effectively manage resource allocation across too many hypervisor instances.

A strong security configuration at the hypervisor layer allows for full protection of resources across the environment, reduced management overhead from a reduction in hypervisor instances, and improved resource allocation.

3.2.2 z/VM built-in security

The z/VM Control Program (CP) provides a basic level of security. CP enforces the definitions that are specified in the system directory for resources, such as minidisks. However, the default security includes the following weaknesses:

- The source for the directory is stored on a CMS minidisk and contains passwords (for users and minidisks) that are stored in clear text
- The default location of the directory source file is somewhat well-known, which leads to the possibility of the passwords becoming known
- Poor granularity of resource management control
- No delegation of security, or separation of authority

Although the use of a directory manager can help to mitigate these points, it cannot eliminate them. Password management is still a concern because a system administrator can still obtain clear-text passwords. Also, the granularity of resource access control is not improved.

These points can be addressed through the use of an External Security Manager (ESM).

3.2.3 Improving z/VM security by using an External Security Manager

An ESM addresses the security points by applying more security configuration and control in addition to (or instead of) standard CP security.

Password management

When an ESM is used, the passwords in the directory are no longer used. User passwords appear in the directory as placeholders, and minidisk passwords are ignored.

An ESM stores passwords in an encrypted form in its database; therefore, the passwords cannot be read from a file. Also, ESMs do not provide a method of retrieving or querying a password.

The use of an ESM also enables the use of *pass phrases* instead of passwords. Like CP directory passwords, ESM passwords are limited to eight characters from a limited character set. Pass phrases (as implemented in RACF) support up to 100 characters from a wider character set.

> **Note:** In the past, some ESMs (including RACF) did not use strong encryption methods to protect passwords; instead, they used “security by obscurity”. RACF now provides the KDFAES password encryption method, which uses much stronger encryption (backed by the hardware CPACF) to protect passwords and pass phrases.
>
> Enabling KDFAES also avoids a previous RACF restriction around pass phrase length. Pass phrases 9 - 13 characters were not supported in RACF without writing an exit to allow them. This restriction is removed when KDFAES is enabled, and pass phrases 9 - 100 characters are supported.

The use of pass phrases instead of passwords is an important consideration that allows z/VM to integrate into a wider enterprise security environment.

Control granularity

Without an ESM, most access controls are broadly applied. For example, anyone who has the read password for a minidisk can access it. To revoke an individual's access to a disk, the password must be changed and then, update any user who still needs access about the new password.

With an ESM in place, access can be granted to individual users or groups of users. More importantly, if access must be revoked from an individual or group, this process can be done without affecting any other valid access. Even if an individual is a member of a group that can access a resource, that individual can be denied access to the resource by defining a specific "deny" entry removing their access.

Delegating security management

When an ESM is not used with z/VM, administration is centralized. It becomes difficult (perhaps impossible) to delegate administration of some user IDs to an alternative manager or administrator. Because all of the definitions of users and their resources is in the user directory, the directory administrator is involved in all changes.

By using an ESM, authority can be delegated to different individuals. For example, RACF provides a capability to make a user ID a security administrator for a group to which the user is connected. This capability is known as *group-special*.

A user with the group-special attribute on their connection to a group can perform security operations against users in that group without having to be made a system-wide administrator. By using this mechanism, different users can be made security administrators of different parts of the z/VM environment (such as one or more Linux system administrators can be given group-special over the Linux systems they manage).

Auditing

The security of a system is only as good as its administrator's ability to prove it is secure. One of the strengths of a z/VM system that is secured by using an ESM is the audit stream that is available. By using the audit records that are created by the ESM, it is possible to document that a security event did (or did not) occur by reviewing the audit trail.

On RACF, auditing is provided by using SMF. To ensure that the audit trail is reliable, it is important to configure SMF to write the audit data out to persistent storage when the SMF data disk fills.

SMF on z/VM RACF provides two data disks for this purpose, and by default switches from one recording disk to the other when one fills. To ensure that it can switch back again when the second disk fills, some processing is required to clean up the data disk and prepare it for its next use.

RACF also allows the system administrator to configure how the system is to respond if both audit data disks ever fills or otherwise become unusable. The default action is for RACF to continue operating without recording audit data. In a production environment, it is essential that this behavior is changed so that no auditable action is allowed to occur without a record being kept. In the `SMF CONTROL` file, RACF can be configured to sever its connection to CP if that SMF data cannot be recorded.

> **Note:** Configuring `SEVER YES` in the `SMF CONTROL` file affects system availability if the SMF data disks fills and SMF data cannot be recorded. The workload continues to operate, but no users can log on nor are any connections to RACF-controlled resources permitted.
>
> Although `SEVER YES` is the recommended configuration, it must not be set until suitable automation or management of the SMF data disks is established.

3.3 Separation of authority

No single individual (or group) should have sole responsibility for establishing, maintaining, configuring, and enforcing the security policy. However, many hypervisor systems are configured in that way; that is, a single ID or privilege level provides universal authority over all aspects of system management. A single focus of administration makes it far too easy for a system to be compromised through the actions of a small number of people, or even an individual.

z/VM can help to separate administrative roles into separate IDs, which makes it difficult for the actions of an individual (accidental or malicious) to create a security event. In addition, z/VM provides a capability that is called *surrogate logon* (known as *logon-by*) to protect VM and other user IDs that are on the system that might be shared for valid administrative purposes.

3.3.1 Surrogate logon: Logon-by capability

Surrogate logon refers to the use of an individual's credentials to gain access to a shared ID. This feature greatly improves the security of shared IDs because the credential of the shared ID is no longer used to access the ID.

Some shared user IDs are provided as part of a z/VM system (including MAINT, MAINTvrm, TCPMAINT, product owning IDs, and service VMs). In addition, Linux guest VMs can be considered shared.

Without surrogate logon, the passwords for these users must be recorded in a way that is accessible by those with an authorized need to access them. A secure way of managing this process is to add complexity, which also might affect recovery time in the case of a problem that required the shared ID to resolve. Done insecurely (such as the "password-on-the-noticeboard"), the shared ID presents a significant risk to the security and integrity of the system.

With surrogate logon in place, authorized system administrators use their own credentials (user ID and password or passphrase) to log on to the shared ID. Credentials are not shared, and only an administrator who is authorized to log on to the shared ID is permitted to do so. Also, when managed by using an ESM, audit records are generated for every successful (or failed) attempt to log on as a shared ID.

IBMVM1 ID

With z/VM 7.1, IBM delivered the IBMVM1 user ID to demonstrate the correct use of the surrogate logon capability. In the default directory that is installed with z/VM 7.1, IBMVM1 is used to access most of the shared IDs on the system.

It might be tempting to use IBMVM1 because it is supplied to log into the system, rather than using it as a model for creating user IDs for system administration staff and correctly setting up surrogate logon. The use of IBMVM1 for anything other than a modeling template represents a significant security exposure for the following reasons:

- The password for IBMVM1 is treated as another shared ID password (for example, as the MAINT password was treated in the past)
- Through surrogate logon, IBMVM1 gives access to many IDs on the system. Therefore, a poorly-secured IBMVM1 password can give access to almost every administration ID that is on a z/VM system.

> **Important:** It is vital to ensure that IBMVM1 is used only as a modeling template for the setup of administration IDs on your system. IBMVM1 must never be used for someone to log on to the system.

After your local system administrator IDs are defined and thoroughly tested to ensure that they function as expected, we recommend that the IBMVM1 ID is disabled by setting the password to `NOLOG`, as described in 15.3, "Using DirMaint to set special passwords for an ID" on page 447 or setting the status to `REVOKED` by using the ESM.

3.3.2 Maintaining separation of administration tasks

All systems today must be designed to mitigate the risk of insider threat. In the case of z/VM, privilege-heavy system administration IDs must be closely managed and monitored.

In older releases of z/VM, the MAINT ID was used for all administration tasks. From user creation and administration to software installation and service, the MAINT ID was used for everything. If a system administrator bothered to create an ID for themselves, it generally was a copy of MAINT with all of its privilege classes copied as well.

As a result, it is common to see many z/VM system administration tasks still being managed through only one or two admin IDs (often the ones supplied with the z/VM system). This issue might represent a security exposure in that one or two user IDs feature high levels of system access authority. If such an ID became compromised, a significant risk to the system can result.

> **Note:** Some tasks are sufficiently complex that they must be done by using the supplied ID; for example, z/VM service by using VMSES/E that uses the MAINT720 ID. Other tasks, such as System SSL certificate maintenance by using the GSKADMIN ID, are done so infrequently that the supplied ID is the most effective one to use.
>
> We are not suggesting that supplied IDs is not be used; rather, these supplied IDs can be used for the purpose they are supplied. We are suggesting that the access and use of these IDs be managed and monitored, used correctly, and not used to circumvent security or management practices.

One of the simplest areas where risk can be reduced is by removing security administration access from the system administration IDs. Security administration is not likely to be a daily task because a system that is established does not need frequent security changes. Also, changes that are needed can often be done automatically by using tools (such as the DirMaint-RACF connector as described in "DirMaint-RACF Connector" on page 321) to mirror directory changes into RACF.

Removing the authority from an administration ID to make security changes make it far less likely that the ID (if it became compromised) can significantly damage a system.

Hint: During the installation or set up of a new system, the convenience of making security changes while performing installation or set up tasks might override the security concerns. However, this convenience must be weighed against the risk to system integrity that might occur if the ID became compromised. Any such convenience must be viewed as temporary, and reviewed and removed at the earliest opportunity.

Separation of authority also can be achieved through automation. In addition to DirMaint-RACF connector, automated actions that are based on the Programmable Operator (PROP) or a tool, such as IBM Operations Manager for z/VM, can be used to enable specific tasks that required escalated privilege to be performed without permanently assigning the higher level of privilege.

3.4 Multifactor authentication

Standard z/VM security is based on the traditional concept of a user ID plus password (and an ESM can replace the password with a passphrase). Multifactor authentication (MFA) enhances standard security by providing an alternative log on flow that can use multiple authentication factors.

The IBM Z Multifactor Authentication V2.1 product (5655-MA1) supports z/VM and z/OS (although a separate installation of the MFA server is needed for z/VM and z/OS systems). The extra factors can be various out-of-band techniques, including RSA SecurID, LDAP, Radius, and one-time password (OTP) systems, such as Google Authenticator.

An example MFA log in to CMS might resemble the following process:

1. Access the MFA server by using a web browser.
2. Authenticate to the MFA web interface by using the authentication factors that are configured in the MFA server for your ID. This process requires entering passwords or authentication tokens according to the factors that are enabled.
3. After successful authentication to MFA, the MFA interface displays an authentication token.
4. In the z/VM log on window, use your standard log on ID with the authentication token that is provided by MFA in lieu of a password or passphrase.

Although MFA is configured across an entire ESM instance, MFA log on is enabled on a per-user basis. Fallback to an ESM password or passphrase can be configured, also on a per-user basis. Also, the attributes of the provided authentication token can be configured, such as the number of uses, the length of time it is valid (from seconds up to 24 hours), and the length and character composition.

Note: When MFA is used for an ID, the ESM password or passphrase is *not* one of the available tokens that can be used. The only time the ESM password or passphrase is used for an MFA-enabled user is if fallback is enabled (and the user uses their fallback).

MFA further enhances the ability of z/VM to integrate into the security policies of an enterprise installation.

3.5 TLS for network traffic

As part of the TCP/IP for z/VM feature, z/VM includes a System SSL component that can provide TLS encryption of network connections. The TN3270 server and FTP server are supported. The IBM Performance Toolkit HTTP interface also can be configured to support TLS.

The z/VM LDAP Server also supports TLS, but it does *not* use the z/VM System SSL. Instead, it uses its own internal TLS implementation.

3.5.1 Why secure z/VM traffic?

For the most part, sensitive data is managed by the Linux instances under a z/VM environment; therefore, securing z/VM traffic might seem to be superfluous. However, it is important to ensure that administrator access and other connections to z/VM are protected from accidental or malicious interception.

TN3270 is the main access method to a z/VM system. Because it is based on the Telnet protocol, it provides no security by default. Therefore, passwords and other information can be retrieved from an intercepted network session.

Such an exposure might occur for entirely innocent reasons. At some point, your organization's network team might be conducting problem identification on the network. As a result, they might be required to capture network traffic. During that traffic capture, you happen to log on to your z/VM system, so your z/VM login details are captured. Then, that capture data might be sent to your organization's network hardware vendor or other external organization for analysis, including your z/VM credentials.

Despite being an EBCDIC data flow, a login by using unencrypted TN3270 can be easily decoded. We used the popular Wireshark tool to trace traffic during a log on to a non-TLS z/VM system, and can easily locate the user ID and password in the stream.

3.5.2 Enabling TLS for z/VM TN3270 server

In this section, we provide an example of how to configure TLS to protect TN3270 communications with z/VM. For this example, we demonstrate how the Ansible system management tool and its OpenSSL modules (running on Linux) can be used to create and manage certificates that secure z/VM communication.

> **Note:** Ansible is a popular system management tool that can be used to simplify the configuration of large system deployments across various platforms. Although no native Ansible modules are available for z/VM, it is still possible to use Ansible for tasks in support of z/VM management.

Ansible provides the following modules that use OpenSSL to create and manage certificates:

`openssl_privatekey` This module is used to create a private key file that can be used in subsequent certificate operations.

`openssl_csr` This module creates Certificate Signing Request (CSR) files, which are sent to a Certificate Authority (CA) to generate a certificate.

`openssl_certificate` This module performs various operations on certificates, including signing (generation of a certificate from a CSR), and verification and validation of certificates.

These are task definition files that can be included as required into Ansible playbooks.

Example 3-1 shows the Ansible task that can be used to create a CA certificate. The variables `ca_key_path`, `ca_csr_path`, and `ca_cert_path` are provided in the playbook that starts the task, or in a variable file (perhaps included by way of the host inventory).

Example 3-1 Ansible task for creation of a Root CA

```
---
- name: generate CA private key
  openssl_privatekey:
    path: "{{ ca_key_path }}/OurCA.pem"

- name: generate CA CSR
  openssl_csr:
    path: "{{ ca_csr_path }}/OurCA.csr"
    privatekey_path: "{{ ca_key_path }}/OurCA.pem"
    common_name: "Our System Root Certificate Authority"
    organization_name: "OurCorp"
    organization_unit_name: "IT LinuxONE"
    country_name: "AU"
    basic_constraints:
      - 'CA:TRUE'

- name: generate CA certificate
  openssl_certificate:
    path: "{{ ca_cert_path }}/OurCA.crt"
    privatekey_path: "{{ ca_key_path }}/OurCA.pem"
    csr_path: "{{ ca_csr_path }}/OurCA.csr"
    provider: selfsigned
```

After the CA is created, it can be used to sign certificates for any purpose, including the creation an intermediate CA, if needed.

The use of Ansible to create a certificate that is signed by our CA is shown in Example 3-2. Here, we are creating the z/VM private key, and a CSR with the details of the certificate we want to generate. The `openssl_certificate` module is used to sign the certificate by using the CA certificate. Finally, a `PKCS#12` file is created that contains the server certificate, server private key, and CA certificate.

Example 3-2 Ansible task for creating a certificate and PKCS#12 package file

```
---
- name: generate z/VM private key
  openssl_privatekey:
    path: "{{ ca_key_path }}/zVM.pem"

- name: generate z/VM CSR
  openssl_csr:
    path: "{{ ca_csr_path }}/zVM.csr"
    privatekey_path: "{{ ca_key_path }}/zVM.pem"
    common_name: "zVM.our.corp"
    organization_name: "OurCorp"
    organizational_unit_name: "IT LinuxONE - z/VM"
    country_name: "AU"
    keyUsage: ["digitalSignature","keyAgreement"]
    extendedKeyUsage: ["clientAuth","serverAuth"]
```

```
- name: generate Certificate
  openssl_certificate:
    path: "{{ ca_cert_path }}/zVM.cert"
    csr_path: "{{ ca_csr_path }}/zVM.csr"
    ownca_path: "{{ ca_cert_path }}/OurCorp.cert"
    ownca_privatekey_path: "{{ ca_key_path }}/OurCorp.pem"
    ownca_not_after: +365d
    provider: ownca

- name: Generate PKCS12 file
  openssl_pkcs12:
    action: export
    path: "{{ ca_cert_path }}/zVM.p12"
    friendly_name: ZVMTLS
    privatekey_path: "{{ ca_key_path }}/zVM.pem"
    certificate_path: "{{ ca_cert_path }}/zVM.crt"
    other_certificates: "{{ ca_cert_path }}/OurCorp.crt"
    passphrase: zvm4secret
```

z/VM System SSL supports the use of a `PKCS#12` key file as an alternative to the traditional gskkyman key database (KDB). Although the KDB format still has some advantages on z/VM (it is easier to have different certificates for different purposes stored in the same key database, for example) the easy generation of a `PKCS#12` file by using common utilities makes it attractive. In our example, after the `PKCS#12` file that is generated by the Ansible task is sent to z/VM, the configuration is almost complete.

We used FTP as a transport to send the file by using a single **curl** command, as shown in Example 3-3. We then also used FTP to store the `PKCS#12` file passphrase in the required stash file (`.p12pw`).

Example 3-3 Transferring the PKCS#12 file to z/VM, and creating the passphrase stash file

```
$ curl -v -T /etc/pki/tls/certs/zVM.p12 -Q "CWD /../VMBFS:VMSYS:GSKSSLDB/"
ftp://maint.by.ibmvm1:ibmzvm@9.60.86.71/
*   Trying 9.60.86.71...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
Connected to 9.60.86.71 (9.60.86.71) port 21 (#0)
< 220-FTPSERVE IBM VM Level 710 at IBMZVM.IBM.COM, 11:38:34 UTC SATURDAY
2020-09-12
< 220 Connection will close if idle for more than 5 minutes.
> USER maint.by.ibmvm1
< 331 Send password please.
> PASS ********
< 230-MAINT logged in; working directory = MAINT 191 (ReadOnly)
< 230 write access currently unavailable
> PWD
< 257 "MAINT.191" is working directory (ReadOnly)
> SYST
* Entry path is 'MAINT.191'
< 215-z/VM Version 7 Release 1.0, service level 2001 (64-bit)
<     VM/CMS Level 29, Service Level 2001
< 215 VM is the operating system of this server.
```

```
> CWD /../VMBFS:VMSYS:GSKSSLDB/
* ftp_perform ends with SECONDARY: 0
< 250 BFS working directory is /../VMBFS:VMSYS:GSKSSLDB/
> EPSV
* Connect data stream passively
< 229 Entering Extended Passive Mode. (|||1067|)
*   Trying 9.60.86.71...
* TCP_NODELAY set
* Connecting to 9.60.86.71 (9.60.86.71) port 1067
* Connected to 9.60.86.71 (9.60.86.71) port 21 (#0)
> TYPE I
< 200 Representation type is IMAGE.
> STOR zVM.p12
< 125 Storing file 'zVM.p12'
} [5934 bytes data]
* We are completely uploaded and fine
* Remembering we are in dir ""
< 250 Transfer completed successfully.
100  5934    0     0  100  5934      0   186k --:--:-- --:--:-- --:--:--  193k
* Connection #0 to host 9.60.86.71 left intact
$ echo zvm4secret | curl -v -B -T - -Q "CWD /../VMBFS:VMSYS:GSKSSLDB/"
ftp://maint.by.ibmvm1:ibmzvm@9.60.86.71/zVM.p12pw
*   Trying 9.60.86.71...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
Connected to 9.60.86.71 (9.60.86.71) port 21 (#0)
< 220-FTPSERVE IBM VM Level 710 at IBMZVM.IBM.COM, 11:38:34 UTC SATURDAY
2020-09-12
< 220 Connection will close if idle for more than 5 minutes.
> USER maint.by.ibmvm1
< 331 Send password please.
> PASS ********
< 230-MAINT logged in; working directory = MAINT 191 (ReadOnly)
< 230 write access currently unavailable
> PWD
< 257 "MAINT.191" is working directory (ReadOnly)
> SYST
* Entry path is 'MAINT.191'
< 215-z/VM Version 7 Release 1.0, service level 2001 (64-bit)
<     VM/CMS Level 29, Service Level 2001
< 215 VM is the operating system of this server.
> CWD /../VMBFS:VMSYS:GSKSSLDB/
* ftp_perform ends with SECONDARY: 0
< 250 BFS working directory is /../VMBFS:VMSYS:GSKSSLDB/
> EPSV
* Connect data stream passively
< 229 Entering Extended Passive Mode. (|||1067|)
*   Trying 9.60.86.71...
* TCP_NODELAY set
* Connecting to 9.60.86.71 (9.60.86.71) port 1067
* Connected to 9.60.86.71 (9.60.86.71) port 21 (#0)
> TYPE I
< 200 Representation type is TEXT.
```

```
> STOR zVM.p12pw
< 125 Storing file 'zVM.p12pw'
} [11 bytes data]
* We are completely uploaded and fine
* Remembering we are in dir ""
< 250 Transfer completed successfully.
100 11 0     0  100 11 0   186k --:--:-- --:--:-- --:--:--  193k
* Connection #0 to host 9.60.86.71 left intact
$
```

Note: If the files are newly created and not replacing files, it might be necessary to modify the permissions of the files uploaded. The GSKADMIN ID can be used for this purpose.

After the `PKCS#12` and passphrase stash files are created, you can update the TCP/IP configuration. The SYSTEM DTCPARMS file is updated to point System SSL to the uploaded `PKCS#12` file (see Example 3-4).

Example 3-4

```
:nick.SSL*     :type.server
               :class.ssl
               :stack.TCPIP
               :Parms.KEYFile /etc/gskadm/zVM.p12
```

Note: For more information about GSKADMIN, see “Encrypting communications by using SSL/TLS on z/VM” on page 221.

4

Optional extra features of z/VM

This chapter contains information about optional products and features that are useful for managing, backing up, and automating a z/VM environment.

It includes the following topics:

4.1 IBM Cloud Infrastructure Center

IBM Cloud Infrastructure Center is an Infrastructure as a Service (IaaS) management solution. It provides on-premises cloud deployment of Linux guests under z/VM and integrates with high-level cloud automation tools; for example, IBM Cloud Automation Manager, VMware vRealize Automation (vRA), and vRealize Orchestration (vRO). It is built with OpenStack compatible APIs, which allows current OpenStack implementations to easily integrate a z/VM environment.

4.1.1 Infrastructure management

Cloud Infrastructure Center provides a consistent experience to manage the virtualization environment (see Figure 4-1). It allows the definition, instantiation, and lifecycle management of virtual infrastructure, image deployment, and resource management.

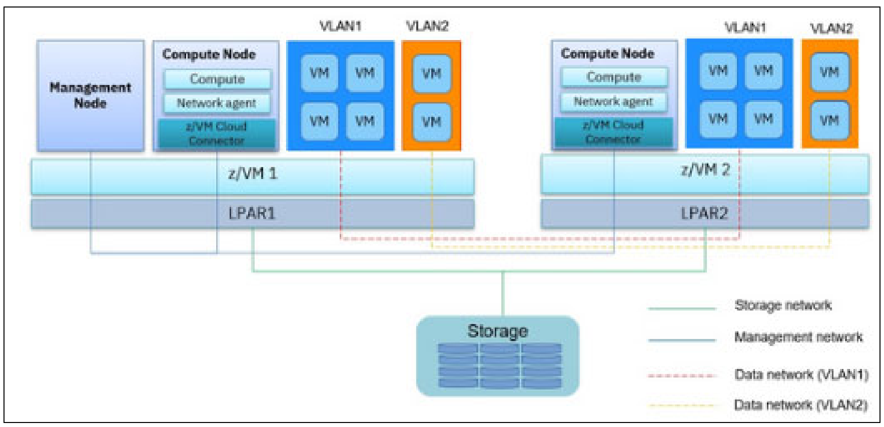


Figure 4-1 IBM Cloud Infrastructure Center Architecture on z/VM

Consider the following points regarding the architecture that is shown in Figure 4-1:

- For one cloud, only one management node must be set up that manages all of the compute nodes.
- For each to-be-managed z/VM, one compute node is required.
- The management node can be installed at the same z/VM with one of the compute nodes, but they must be on different Linux virtual machines of that z/VM.

4.1.2 Automation

By using a library of predefined images, it is possible to quickly change the environment to better fit any situation. The workloads also can be better fit by starting new web servers, or shutting down unused back-end servers.

The API can be used to check key metrics from the environment, and not only from the deployed guests. With those interfaces, it is possible to verify resource usage, software versions, service status, application logs, and more.

4.1.3 Integration

Cloud Infrastructure Center provides RESTful APIs to integrate with industry-standard management tools. Its built-in OpenStack compatible APIs allow an integrated, vendor-agnostic IaaS solution. Therefore, you do not need any extra skills to easily integrate a z/VM environment on the cloud solution that is in place.

For more information, see this web page.

4.2 OpenShift

Container-based applications provide organizations with a new paradigm for application development and deployment. Kubernetes and containerized applications allow organizations to streamline their DevOps environment and rapidly deploy, develop, and maintain stateless and stateful applications.

Red Hat OpenShift Container Platform (RHOCP) is Red Hat's enterprise Kubernetes distribution. It provides developers, IT organizations, and business leaders with a hybrid cloud application platform for developing and deploying new containerized applications on a secure platform.

RHOCP also offers scalable resources that require minimal configuration and management. Developer can create and deploy applications by delivering a consistent environment throughout the lifecycle of the application, including development, deployment, and maintenance. The self-service capabilities and minimal maintenance requirements reduce the burden on IT organizations that are associated with deploying and maintaining application platforms.

4.2.1 Benefits of Red Hat OpenShift Container Platform

RHOCP includes the following benefits:

- Faster innovation and time-to-value: Well-defined recommendations and best practices for designing and operating RHOCP lead to a standardized architecture.
- Simplification: A consistent set of APIs is defined for developers and administrators. Kubernetes Operators dramatically simplify a solution architecture and the required skills to operate it.
- Safer deployments: RHOCP automates and secures the orchestration and life-cycle management of applications within its cluster.
- Pre-validated patterns: Best practices for the RHOCP solution stack are validated by Red Hat and IBM, which results in a highly prescriptive and future-oriented solution.
- Cost efficient: RHOCP is based on open source technologies that are fully supported by Red Hat.

4.2.2 Benefits of RHOCP on IBM Z and IBM LinuxONE

Specially, when running RHOCP on IBM Z and IBM LinuxONE, customers can take full advantage of the following platform capabilities and characteristics:

- Non-disruptive growth, which enables vertical and horizontal scalability to help accommodate substantial increases in workload on-demand.
- Highest scalability for millions of containers and thousands of Linux guests in one physical machine have been proven
- Containerized workload within a state of the art microservices architecture, while accessing traditional transactional z/OS services and databases.
- Take advantage of advanced security and confidential cloud computing, including FIPS 140-2 Level 4 certification with the IBM Z Cryptographic capabilities.
- Multi tenancy with full LPAR isolation (EAL5+) allows administrators to share a single hardware securely. Even virtual machines on the same hardware offer EAL4 certification.

4.2.3 Typical RHOCP deployments and Topologies

Based on the use case, you can have different deployment and topologies with RHOCP:

- Deploying RHOCP as the only cloud environment on IBM Z and IBM LinuxONE.
- Deploying RHOCP with another, non-containerized workload that you also run as back-end in a z/VM environment with Linux servers on IBM Z and IBM LinuxONE hardware.
- Deploying RHOCP in colocation with a traditional operating and transactional environment and services, such as z/OS, z/VSE, or z/TPF on IBM Z hardware.

4.2.4 Production environment

Because of a design with High Availability in mind, an RHOCP cluster must be deployed in a production environment with three control plane nodes or control planes that are spread across multiple LPARs in one or multiple physical machines.

RHOCP provides powerful orchestration capabilities for the container workload that you are deploying. For orchestration to occur seamlessly, it is essential to provide sufficient resources to your cluster.

For a production environment, it is highly recommended to provide at least the preferred or recommended hardware and system requirements that are shown in Figure 4-2.

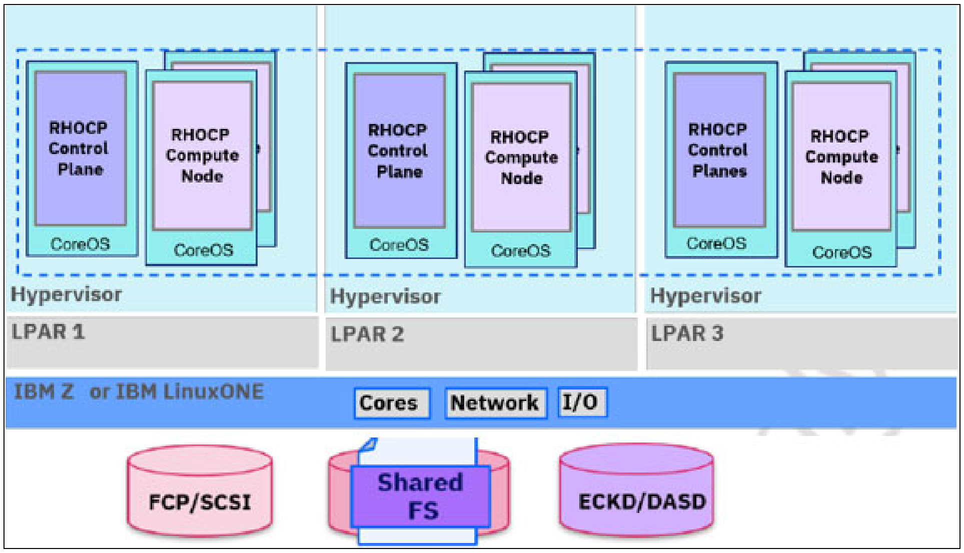


Figure 4-2 Production like RHOCP Cluster in z/VM (hypervisor) environment

Minimum requirements

The following minimum requirements must be met:

- Hardware:
  - Three LPARs with six IFLs each that support SMT2
  - OSA or RoCE network adapters in HA configuration
  - HiperSockets, which are attached to a node
  - In a z/VM environment, the HiperSockets network can use the z/VM VSWITCH bridge to enable the communication to a z/VM VSWITCH
- z/VM hypervisor instances:
  - Three guest virtual machines for RHOCP control plane machines, one per hypervisor
  - At least six guest virtual machines for RHOCP compute machines, distributed across the hypervisor instances
  - One guest virtual machine for the temporary RHOCP bootstrap machine
- Disk storage for the z/VM hypervisor guest virtual machines:
  - FICON attached disk storage (DASDs).

    For z/VM hypervisor, these can be z/VM minidisks, fullpack minidisks, or dedicated DASDs. To reach the minimum required DASD size for Red Hat Enterprise Linux CoreOS (RHCOS) installations, you need extended address volumes (EAVs) in z/VM. It is highly recommended to use Parallel Access Volume access (HyperPAV) and High-Performance FICON (zHPF) to ensure optimal performance.
  - FCP attached disk storage.
  - For FCP/SCSI attached storage.
- Storage and main memory:
  - 16 GB for RHOCP control plane machines
  - 8 GB for RHOCP compute machines
  - 16 GB for the temporary RHOCP bootstrap machine

For the operating system, instances of z/VM hypervisor must be included on each LPAR.

> **Note:** It is also suggested that production systems are carefully monitor in terms of performance and resources consumption and make sure that RHOCP does not encounter bottlenecks in available hardware.

4.2.5 Virtualization and hypervisors

RHOCP on IBM Z and IBM LinuxONE is deployed in a virtualized environment and offers the following configuration options:

- Using z/VM as hypervisor

  The most common virtualization of RHOCP on IBM Z and IBM LinuxONE takes advantage of the z/VM hypervisor. This virtualization offers the security certification EAL 4+ which is the highest level of certification and represents the highest level of isolation in virtualized environments. Because z/VM is a well-established hypervisor for the mainframe, it is a proven technology and commonly used by many customers.

- Using a KVM-based deployment

  An alternative approach for virtualization is planned to be based on a KVM-based deployment.

  > **Note:** This option is considered for one of the upcoming releases.

4.3 Operations Manager

The Operations Manager product improves the management of z/VM systems by automating tasks, responses, and sessions (see Figure 4-3 on page 89). It also monitors thresholds and messages. It improves accuracy, reduces operational effort, and improves administrative notification.

It provides the following main functions:

- Task automation

  By using a schedule, periodic commands can be run without operator intervention. For example, it can run a task to clear console files on the spool, erase DIRMAINT log files, or send PERFSVM diagnostic data by email for external processing.

- Response automation

  By using a set of rules, message can be matched to a pattern, and different commands can be run (depending on the received message). For example, it can take a memory dump from a Linux server when it enters a Disabled Wait state, or automatically use Live Guest Relocation to move Linux guests to another SSI member during a maintenance window.

- Monitoring

  Operations Manager can monitor spool usage, page space, system events, and user consoles, which enables multiple users to read the same user console. It also can send alerts when thresholds are met, so if the spool usage is above the configured threshold, it can send an email alerting the system administrators to correct the issue before the spool is full.

- Session automation

  Operations Manager can automatically log on to a user ID, run a set of commands, and shut down that user ID after finishing.

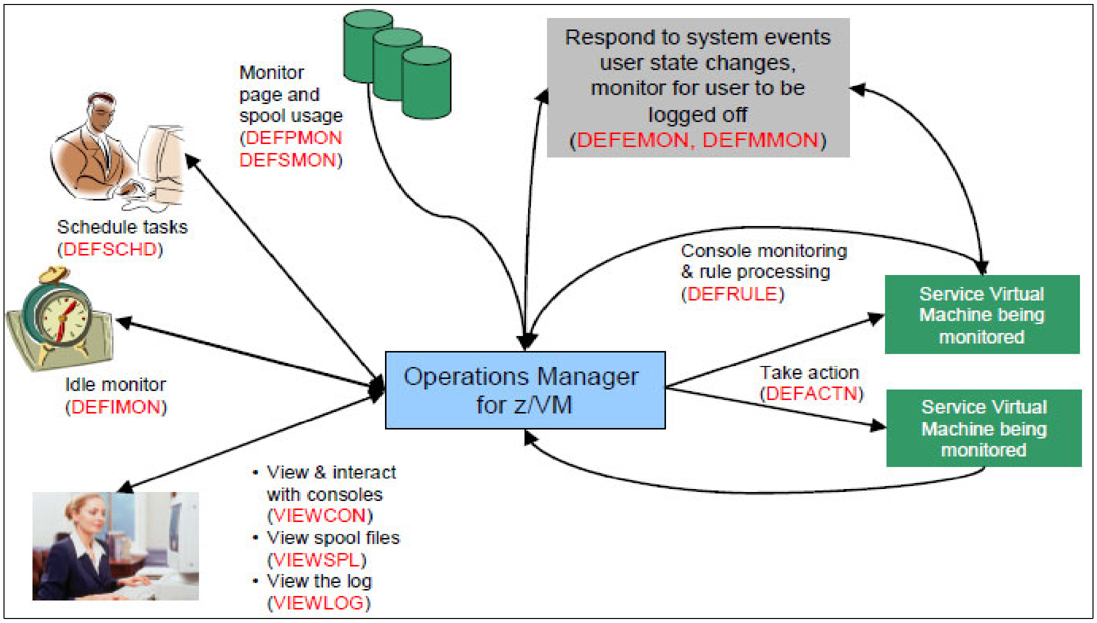


Figure 4-3 z/VM Operations Manger relationship

By using Operations Manager, routine tasks can be automated and the need for manual running of those tasks can be reduced, which removes the possibility of human error, maintains repeatability, and allows for timely execution.

It also reduces operational requirements by allowing the automated handling of messages on common situations. It removes the need of operational personal to interpret the messages and to take the correct action. By using pattern matching, Operations Manager can take actions immediately, without the need of a human operator.

It provides fast action and fast notifications. By monitoring system metrics and user logs, Operations Manager can take action whenever any message is received or any metric meets a predefined threshold. On events that require manual intervention, Operations Manager can send alerts with no delay.

4.4 Backup and Restore Manager

A system backup is required to reduce the downtime that results from hardware failures or operational mistakes. By using Backup and Restore Manager for z/VM, different types of backups can be performed, depending on the operational needs. By running inside z/VM, it allows for backups to be taken while the system is running, without the need to shut down the system.

A replica is *not* a backup. A replica can protect against hardware failures, but not operational failures. A file that was saved with incorrect data is replicated on the Disaster Recovery site, and both sites have a stored corrupted version. With a backup implementation, several versions of the file exist, which allows system administrators to recover the newest good version of any backed-up file.

Backup and Restore Manager can generate file-level backups for individual files, or a group of files. This feature is important to keep a safe copy of critical system configuration files, such as `SYSTEM CONFIG`, `USER DIRECT`, log files, automation scripts, and other files (see Figure 4-4).

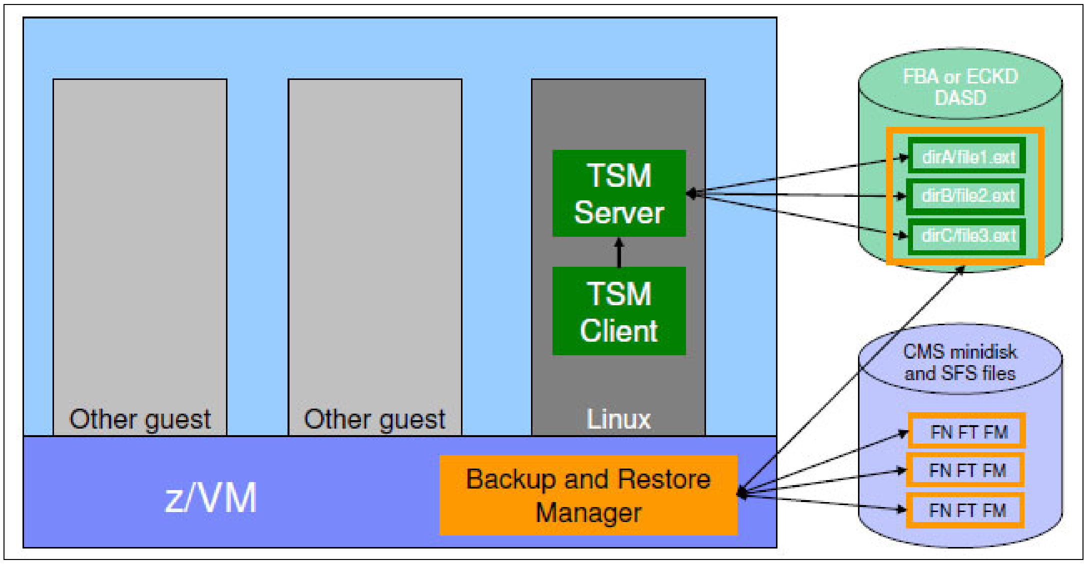


Figure 4-4 z/VM Backup and Restore running on z/VM environment

It can create full disk backups from SFS, FBA, CKD, or CMS minidisk. This feature is useful for backing up a set of Linux servers, for example. It is helpful to create a system backup before applying a set of fixes to allow for rapid recovery if an update fails.

Backups can be a full backup or an incremental backup, and stored on tape, CMS minidisk, or a Shared File System (SFS) volume. Backups can be compressed to reduce storage requirements, and have a limited number of versions, with automatic aging and pruning of backup sets. To protect the backup, you can encrypt the backup by using your own routines, third-party routines, or encryption-enabled tape devices.

The restore process can be run from a full screen interface, or the command-line interface. It allows users to restore their own data so that files that are deleted or corrupted can be restored without administrator intervention. It also enables system administrators to restore any backed-up data.

4.5 CMS Pipelines and VM utilities

In this section, we describe CMS pipelines and VM utilities that are available.

4.5.1 CMS Pipelines

CMS Pipelines is the implementation of the pipeline concept under CMS. It allows programs on the pipeline to read records, process them, and write them to a standard interface. Programs in the pipeline, also called *stages*, can be combined to run various functions. They are included on z/VM.

The pipeline module contains a library of stages to perform a handful of utility functions, such as reading and writing files, sorting and changing records, and even opening TCP sockets. Those functions can be used to perform various processing without having to write the filters and subroutines. It is even possible to use REXX to create a custom stage if no suitable stages are available.

Every stage consists of a program, its arguments, and the pipeline separator (the vertical bar "|" by default). The programs are not common CMS programs, but are created specially to be run by the pipeline.

Every stage reads one line of data from the pipeline, processes it in some way, and writes it back to the pipeline. Every stage is independent from the others; they do not have to know the previous or following stage. Data flows to the next stage when the current stage processed it.

This process gives the pipeline a robust behavior. Consider the following points:

- The pipeline starts only if every parameter on every stage is valid and all programs can be allocated on memory.
- A syntax error in any stage or an error on any program on the pipeline stops the entire pipeline.
- Data is read again from input only when the previous record is processed and its output is sent to the next stage.

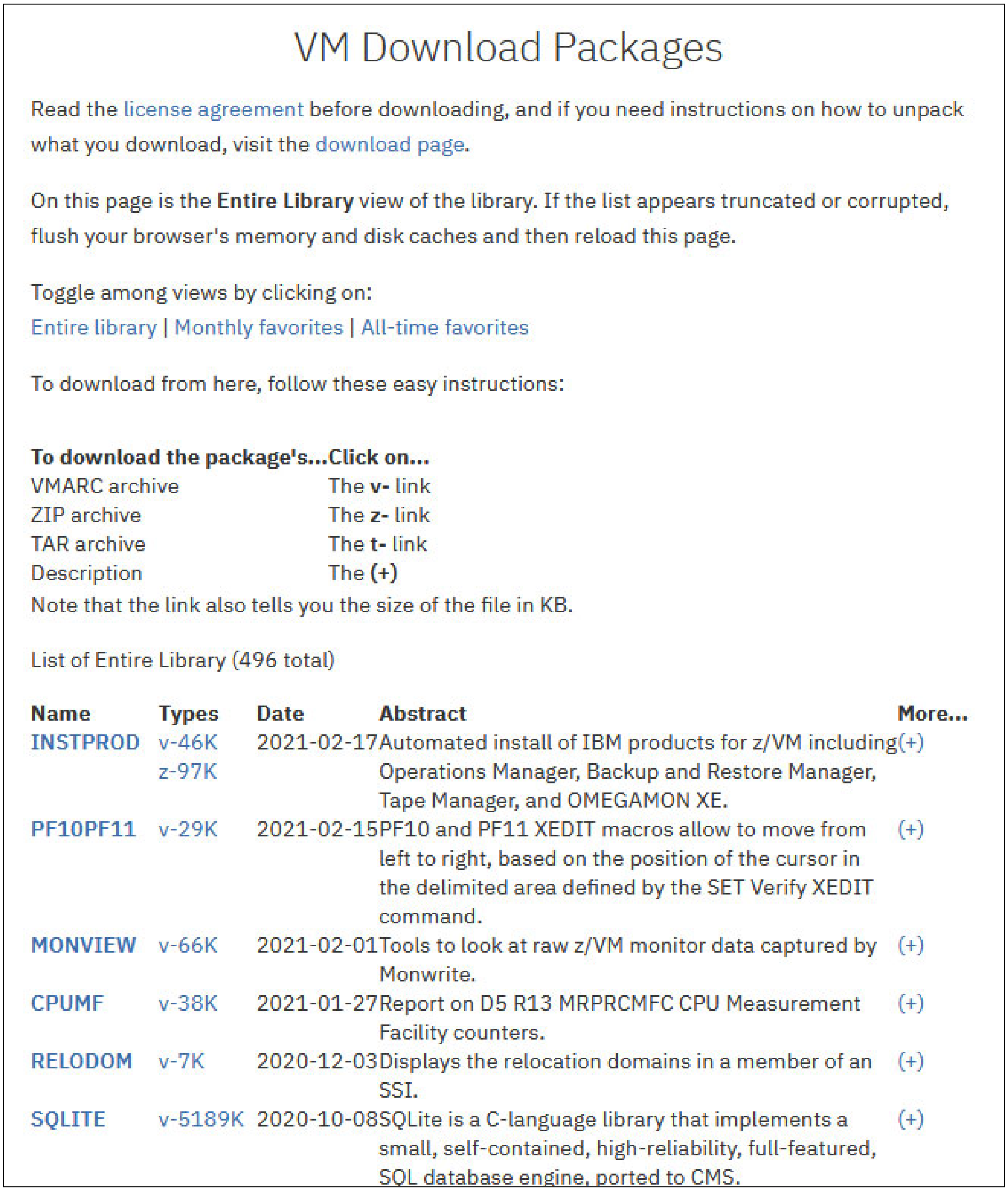
The **HELP PIPE** command contains more information about the available stages, with samples about the usage of them.

4.5.2 VM Utilities

VM Utilities is a repository of tools for z/VM administration. It includes several programs to implement functions that are not available on z/VM. They are intended for system administrators, system programmers, and support personal.

The utilities are available for download for no charge at this web page. Even if those utilities are hosted by IBM, they are not supported in any way. They are provided as is, without any type of warranty.

The available packages range from special macros for XEDIT to the implementation of SQLITE on CMS, enhancements for DDR, and parsers for MONITOR data, among others. Some packages list another package as a dependency; therefore, read the documentation for more information about the required downloads (see Figure 4-5).



VM Download Packages

Read the license agreement before downloading, and if you need instructions on how to unpack what you download, visit the download page.

On this page is the **Entire Library** view of the library. If the list appears truncated or corrupted, flush your browser's memory and disk caches and then reload this page.

Toggle among views by clicking on:
Entire library | Monthly favorites | All-time favorites

To download from here, follow these easy instructions:

To download the package's...	Click on...
VMARC archive	The **v-** link
ZIP archive	The **z-** link
TAR archive	The **t-** link
Description	The **(+)**

Note that the link also tells you the size of the file in KB.

List of Entire Library (496 total)

Name	Types	Date	Abstract	More...
INSTPROD	v-46K z-97K	2021-02-17	Automated install of IBM products for z/VM including Operations Manager, Backup and Restore Manager, Tape Manager, and OMEGAMON XE.	(+)
PF10PF11	v-29K	2021-02-15	PF10 and PF11 XEDIT macros allow to move from left to right, based on the position of the cursor in the delimited area defined by the SET Verify XEDIT command.	(+)
MONVIEW	v-66K	2021-02-01	Tools to look at raw z/VM monitor data captured by Monwrite.	(+)
CPUMF	v-38K	2021-01-27	Report on D5 R13 MRPRCMFC CPU Measurement Facility counters.	(+)
RELODOM	v-7K	2020-12-03	Displays the relocation domains in a member of an SSI.	(+)
SQLITE	v-5189K	2020-10-08	SQLite is a C-language library that implements a small, self-contained, high-reliability, full-featured, SQL database engine, ported to CMS.	(+)

Figure 4-5 VM Utilities download website

4.6 zSecure Manager for RACF z/VM

zSecure Manager for RACF z/VM is part of the IBM Security™ zSecure suite. It improves security, allows for efficient administration, and decreases the operational effort for the mainframe security environment.

It integrates with RACF z/VM to simplify complex and time-consuming tasks, allows for mass-changes to be easily run, and helps automate daily tasks, such as the following examples:

- Add or delete user IDs and groups
- Modify access permissions
- Set and revoke user IDs and passwords
- Search for users and groups
- Generate audit reports

zSecure analyses and audits the RACF database and audit logs, and displays security issues, such as missing or invalid definitions, privileged access by users, and others. It also includes support for Linux on z audit events, which helps detect Linux security issues and generate compliance records.

It also analyses the system to help reveal breaches in the system integrity, unintended exposures, potential threats, and other irregularities. A careful analysis generates a report for those exposures, indicates the severity of each, and includes a description that help determines the corrective actions to be taken.

For more information about zSecure, see this IBM Documentation web page.

Part 2

Installation, configuration, and service

In this part, we describe how to install and configure z/VM. We also discuss live guest relocation and how to service z/VM.

We also describe Centralized Service Management (z/VM CSM), which allows you to manage distinct levels of service for a specific group of z/VM systems.

This part includes the following chapters:

5

Installing z/VM

This chapter describes the following z/VM 7.2 installation types:

- Two-node VM Single System Image (VMSSI) cluster
- Stand-alone traditional newly installed system
- Stand-alone upgrade-in-place system

For all of these methods, we describe performing the initial configuration, hardening, and enabling basic system automation.

For more information about how to choose which type of installation based on prerequisites, available resources, and other factors, see 2.2, “Choosing a z/VM installation method” on page 36.

Important: Prevent unnecessary problems and rework by fully completing all tasks in this volume in the order they are presented. Each chapter relies upon actions that are taken in the previous chapter. Also, each chapter and the sections within are sequenced to give you the quickest results.

This chapter includes the following topics:

5.1 Obtaining z/VM through electronic download

z/VM can be ordered and delivered electronically through IBM Shopz. A detailed description is outside the scope of this book; however, short steps are documented. The steps and links might change over time, but the basic process will remain the same.

You can download the z/VM product installer to a staging machine, such as a workstation, as we did in this example, and later upload them to a File Transfer Program (FTP) server. However, you can also download them directly to the machine that will be the FTP server, such as a Linux personal computer if it has access to the Internet.

5.1.1 Placing the order

To order z/VM, complete the following steps:

1. Visit the z/VM Buy/Order page at the following URL:

   http://www.ibm.com/vm/buy/

   – You might want to review the links that describe the System Delivery Offering (SDO) and Subscription and Support (S&S) so you understand the details.

   – When you are ready to proceed, under the heading How to Buy, click **IBM Shopz.**

2. The Shopz landing page opens, in which information on Shopz and the product catalog are available. Click **Sign in/Register**.
3. If you do not have an IBMid, which is used to sign in to IBM.com, click **Create an IBMid**. If you have an IBMid, sign in.
4. If you have not used Shopz, choose the suitable registration link to become an enrolled user. If you are registered, choose the sign-in link that applies for you.
5. Click **Create new software orders** for service or products:

   a. Select **z/VM - Products** and choose **VM SDO version 7** in the drop-down menu to the right.

   – Click **Continue**.

6. You can choose to accept the Order Name as-is, or override it with a new name if wanted. We changed our order name to “Products - z/VM Version 720 - 2020-09-21” (see Example 5-1). Click **Continue**.

   Example 5-1 Specify order basics

```
Review and specify the basic details of your order.

Order name: Products - z/VM Version 720 - 2020-09-21
Customer S015xxxxxx - IBM CORP(US)
Operating environment z/VM
Package category Products
Package type [Help] VM SDO version 7
```

7. Select a hardware system on which you plan to run z/VM from the list of hardware systems for your customer number, and click **Continue**.

8. Make the following selections:
   a. For the Group, select **VM - VM All**.
   b. For language, select **English** because z/VM is available in English only.
   c. For the Filter, select **Show all products**.
   d. Click **Show catalog**. A submenu opens.
9. If you intend to create the environment that is described in this book, select the programs that are shown in Table 5-1 from the list of products that displays.

Table 5-1 Programs to select for ordering

Product	Description	Version
5741-A09	z/VM V7 System Image DVD 7.2	7.02.00
5741-A09	Dir Maint Fac Feature 7.2	7.02.00
5741-A09	RACF Security Server z/VM 7.2	7.02.00
5741-A09	Performance Toolkit z/VM 7.2	7.02.00
5654-260	EREP VM	3.05.00
5655-T13	zSecure Mgr for RACF z/VM	1.11.02
5698-IS2	Infrastructure Ste z/VM	1.01.00
5684-043	ISPF Version 3 for VM/SP	3.02.00
(Optional)	RSCS Feature z/VM 7.2	
(Optional)	DFSFS z/VM Primary 7.2	

   If you know which programs you need and did licensed them, you also can select **RSCS Feature for z/VM 7.2 and DFSMS/zVM Primary 7.2**.

   Click **Continue.**

   Important: During the time in which z/VM 7.1 and 7.2 can be ordered, be sure you select the correct release from the list by verifying the Version column.

10. Verify the order and click **Continue.**
11. Verify the entitlements and select any options that are not yet selected. Then, click **Continue.**
12. For the Preferred media, select **Internet** and complete any other fields that might be applicable for your situation, such as Purchase Order, Notify Email, and so on. Then, click **Continue**.
13. Review what you submitted carefully for accuracy. This is your only chance to correct any mistakes.
14. If everything looks correct, click **Submit.**

Order processing

It takes some time for the order to be prepared. You can view the order status anytime through ShopZ. Figure 5-1 shows and the status page for our example order.

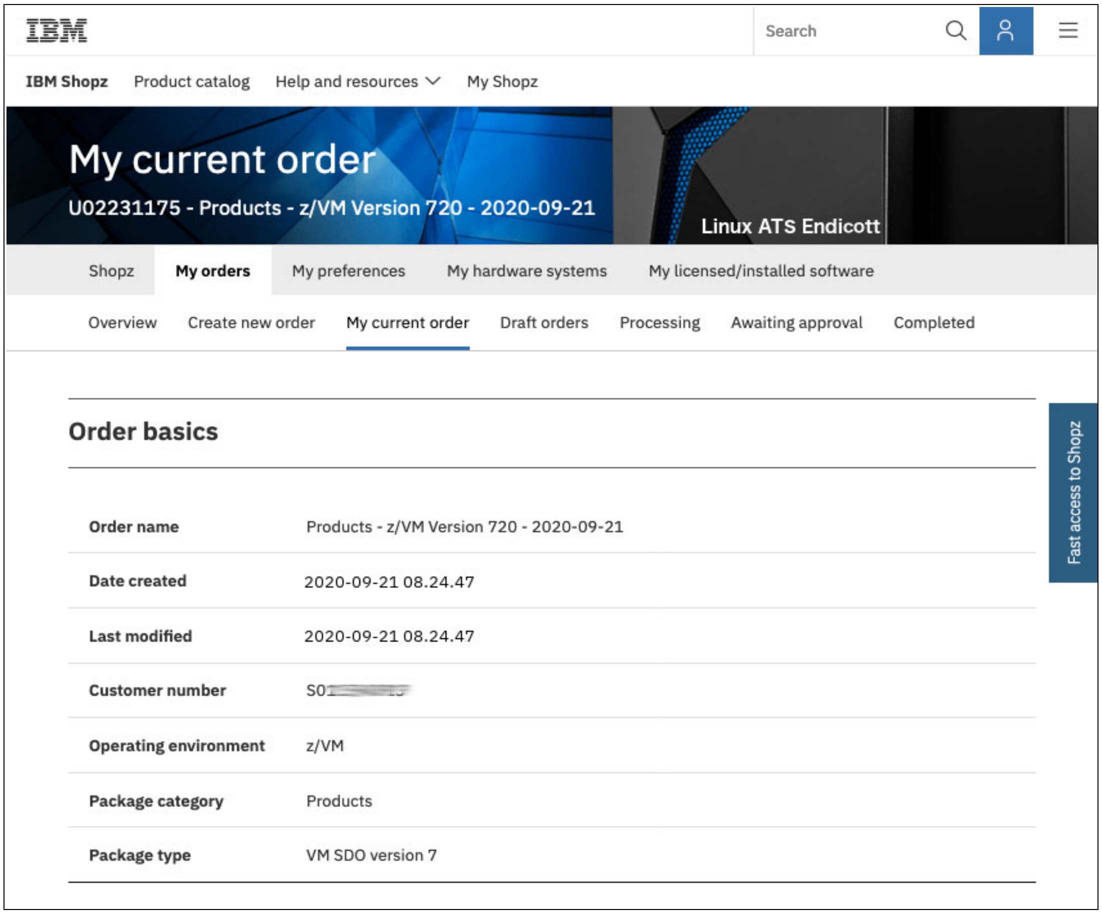


Figure 5-1 The order status page for our example order used to author this book

If this is the first time you are ordering any of the optional features that require a license, the order must go to the entitlements department first to generate the license. Although generated within one business day, license generation might take up to two full business days in some situations.

After the license is generated, the order then moves on to the CSO department for fulfillment. In our example, the email that indicates that the order is ready for download was received after approximately four hours. An example email is shown in Example 5-2.

Example 5-2 Example email that indicates an order is ready for download

```
From:    Software_eDelivery <efactory@us.ibm.com>
To:      Paul W Novak/Endicott/IBM <pwnovak@us.ibm.com>
Date:    2020-09-21 12:18:43
Subject: IBM Order 1234567890 is ready for download.
========================================================================

To >>      pwnovak@us.ibm.com
From >>    efactory@us.ibm.com
Subject >> IBM Order 1234567890 is ready for download.
```

```
ORDER REFERENCE INFORMATION
   IBM customer number:                       S012345678
   PRODUCT: IBM order number:                 1234567890
            ShopzSeries reference number:     U987654321

Refer to the IBM order number when contacting IBM support:
http://www.ibm.com/support

Your order will be available for download on the IBM software
delivery server through "30 Oct 2020".

To access your order directly, go to:
https://www.ibm.com/software/shopzseries/ShopzSeries_public.wss?
action=download&orderId=U987654321
***********************************************************************
*     NOTE: this e-mail is generated by the IBM Corporation           *
*     e-Factory Software, which is processing your order quoted       *
*     in the subject line above on behalf of the IBM entity which     *
*     is supplying you with the software that you have ordered,       *
*     as well as the associated right to use that software.           *
***********************************************************************


IMPORTANT PLEASE READ PRIOR TO ACCESSING THE ABOVE LINK


Before you access the ShopzSeries Download link above, please read
through these steps to familiarize yourself with the DOWNLOAD screen.
Below is a description of what you will be presented with when
receiving your order electronically.  You may receive part of your
order by mail if some components of your order are not available by
electronic delivery.  For any installable items that you received by
mail, follow the Program Directory for z/VM System Delivery Offering
(SDO).


STEP 1:

  Order Packing List - This link contains a media report which lists
all Products contained in your order.  Review this list to ensure
the order is correct.

STEP 2:

  Installation instructions - This link takes you to the instructions
to upload the product material (Product Envelope) to a VM host system
and to prepare for installation. It is recommended that you read or
print these installation instructions in their entirety prior to
downloading the products.  Once completed, these instructions will
direct you to the Program Directory for z/VM Service Delivery Offering
for the complete installation instructions.
```

```
STEP 3:

  VM product material - This link will download the product material to
your workstation using either the IBM Download Director or HTTPS.

STEP 4:

  Product Publications - This link takes you to the online publications
for your specific order.  The number of publications is unique for
each order. You may receive some publications hardcopy because they are
not available online.

STEP 5:

  Additional Publications - This link will point you to the specific
Program Directories for each of the products you ordered.  The Program
Directory for z/VM System Delivery Offering (SDO) is required for all
installations.

Depending on what was ordered, you may receive some of the content in
a .zip format.

1. Download the xxxxxxx.zip file to your workstation.
2. Extract the file(s) using an unzip function.

In most cases the extracted file(s) are usable directly after they are
extracted.

If your order contained a product that was originally packaged on a
CD-ROM, this link may also point to .iso files.
If your order did NOT contain any CD-ROM products, skip the rest of this
step, you are now ready to access your order.  Click on the ShopzSeries
link to get to the Download page and proceed through the steps you just
reviewed.

However, in some cases your order may contain ADDITIONAL MATERIALS or
ADDITIONAL PUBLICATIONS that were originally packaged on CDs.  These
may be provided as ISO9660 images with a file extension of .iso.
An ISO0660 CD-ROM image is a single large file that is an exact
representation of the data and programs as the appear on a CD,
reflecting both the content and the logical format.

To use .iso files, you have to options:
 - Option 1. Create a physical CD. This requires that your workstation
has CD-write capability and software that supports ISO9660 format.
When you create the physical CD, this is an exact copy of the original
CD and has all of the characteristics of the original image (for
example, special file names, and if applicable being a bootable CD).

 - Option 2.  Use virtual CD software. Virtual CD software emulates
your computer's CD-ROM drive, enabling you to execute programs, view,
and use the data provided in the CD image directories and files. This
is an alternative to creating a physical CD.  This software must
support .iso files.
```

```
Read licenses and follow the procedures specific to any software that
you use to process these packages.


You are now ready to access your order.  Click on the ShopzSeries link
to get to the download page and proceed through the steps you just
reviewed.

***********************************************************************
*     Please do not reply to this email, as                           *
*     it was sent by an unattended machine.                           *
***********************************************************************
```

When you receive the email, it includes the URL for downloading your order. Use a web browser to go to that URL.

Important: Be sure that you obtain the code in your order in a timely manner. In Example 5-2 on page 100, it specifies a date through which the order is available:

```
Your order is available for download on the IBM software delivery server
through "30 Oct 2020.
```

After the date passes, the code is no longer available and you must go through the entire ordering process again if you did not download everything.

Order retrieval

After you clicked the delivery URL in the email, you must review several sections of the order retrieval web page, as shown in Figure 5-2.

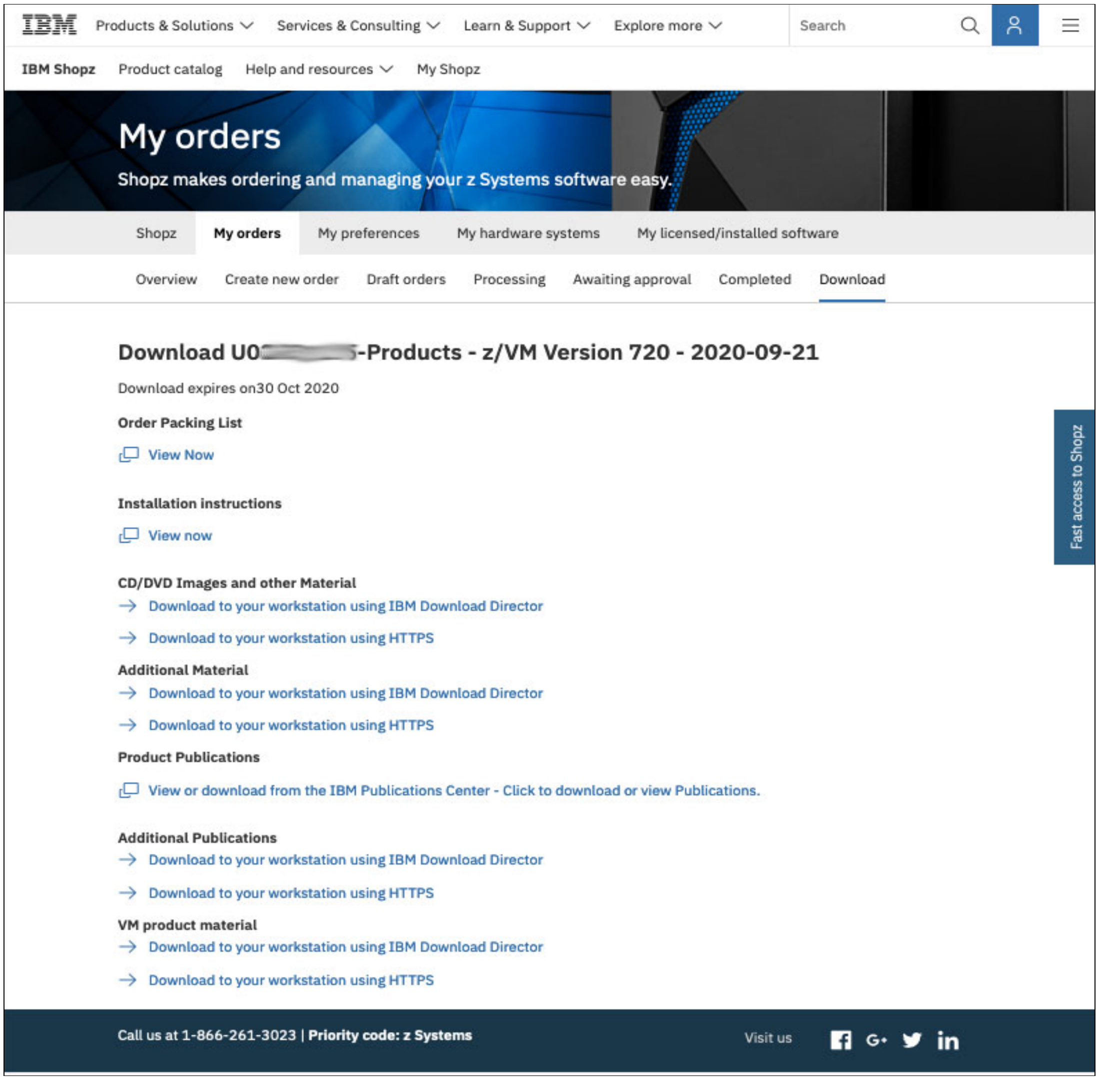


Figure 5-2 Example web page to download the order from ShopZ

This web page features the following sections:

- Order Packing List: A PDF document that contains the list of available products and manuals that you ordered.

  Always open are review this PDF first. Confirm that it correctly lists everything you ordered and that you did not mistakenly forget to order an item.

  We recommend that you download this file and save it for future reference. It should be saved wherever you store documents for permanent reference.

- Installation Instructions: Clicking **View now** takes you to this product installation web page.
- CD/DVD Images and other Material
- Additional Material
- Product Publications

- Additional Publications: Download a z/VM SDO document (four pages).
- VM product material

Complete the following steps:

1. Download any publications or documents first. Save them all to a common location so that you can easily locate them when needed.

 > **Note:** In the example that is used in this book, we chose the **Download to your workstation using IBM Download Director** option \ whenever it was available. Download Director is a multi-connection threaded applet that results in a much faster download than the basic HTTPS method.

2. Select the products that you want to download, as shown in Figure 5-3.

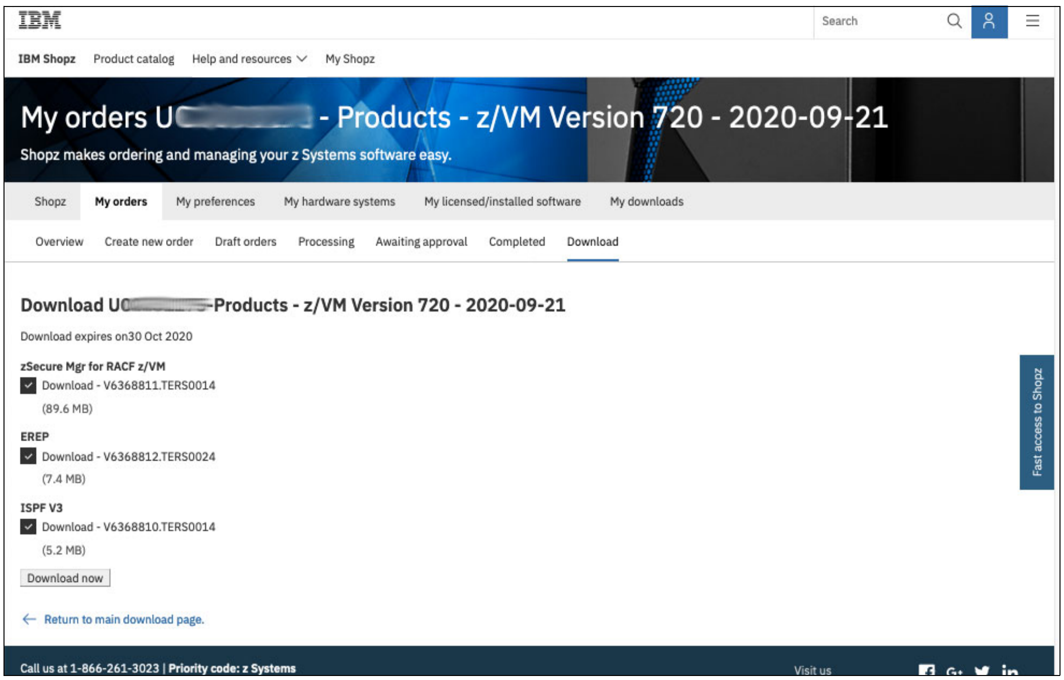


Figure 5-3 Download Products

You are presented with the list of files that correspond to the products that you want to download, as shown in Figure 5-4.

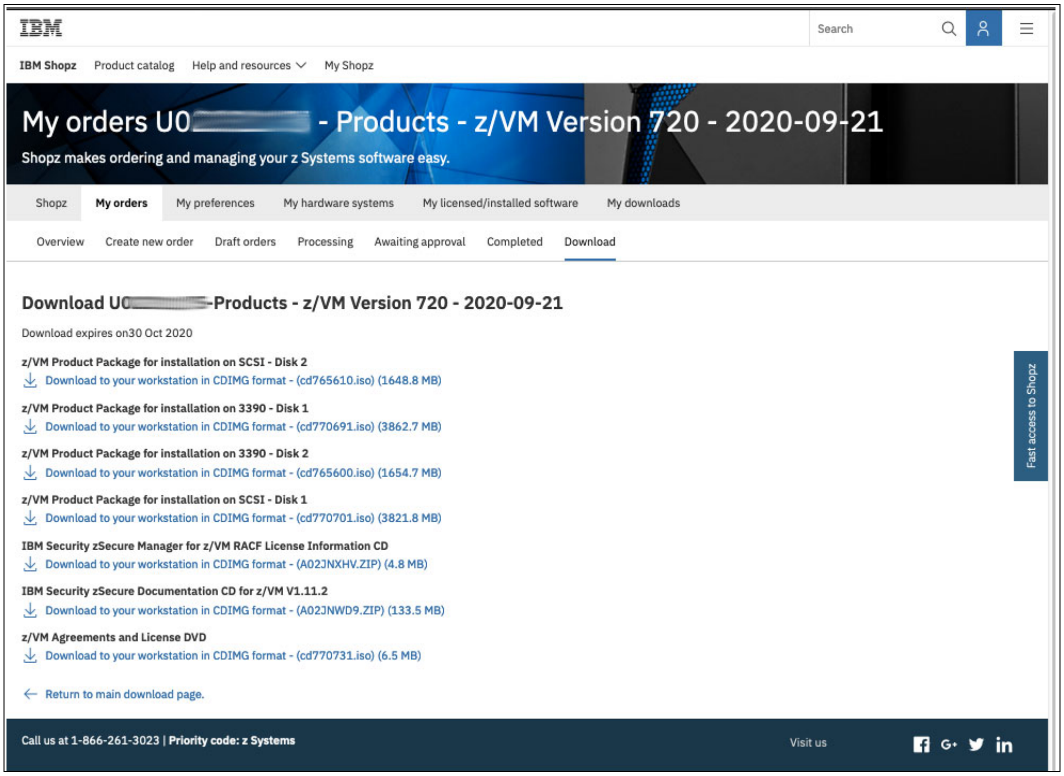


Figure 5-4 Files that are associated with selected package downloads

The window that is shown in Figure 5-5 opens.

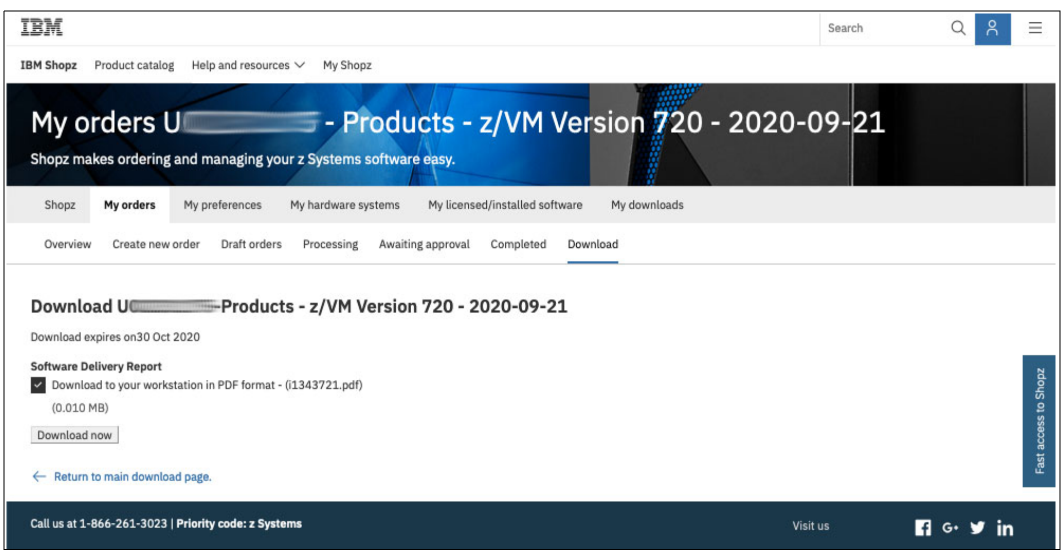


Figure 5-5 Downloads confirmation window

3. Click **Download now** to display the window that is shown in Figure 5-6. The options were selected because z/VM 7.2 is installed onto 3390 DASD. The 1.9 GB of data was downloaded relatively quickly because of multiple connections that are opened by using IBM Download Director.

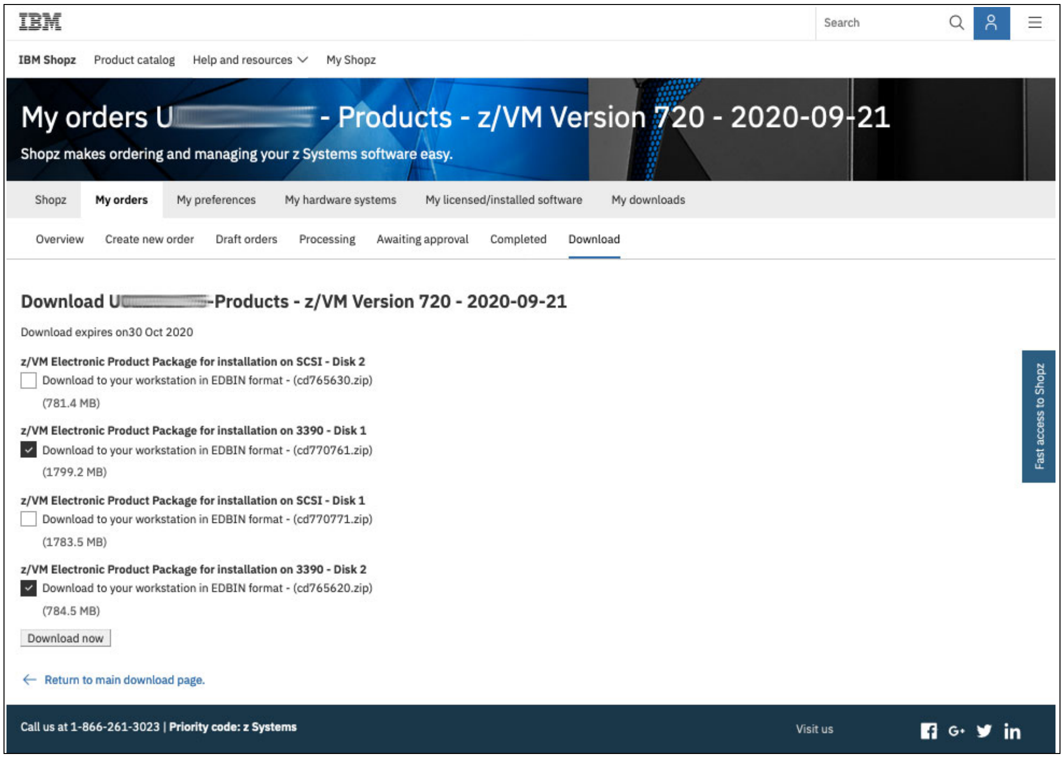


Figure 5-6 Selecting products to download

In our example environment, all of the items that were downloaded from the Shop Z order are uploaded to a Linux-based FTP server and unpacked in the next section. The file was staged on a Linux workstation. If you plan to use an FTP server for installing z/VM, you might want to download directly to that server.

The files are displayed from the shell:

```
/tmp/vmisodisk/
bash [3517]===> ls -alph cd760530.zip
-rw-rw-r-- 1 pwnovak atsc9c 1.6G Feb  5 11:35 cd760530.zip
```

The download of the z/VM installation compressed file is complete. In addition to reviewing the Installation Instructions page, it is recommended that you also review the VM Installation Tips web page.

You can now configure an FTP server as described in the next section.

5.2 Configuring an FTP server for z/VM installation

> **Important:** Do not unpack or expand any of the materials that are downloaded from Shop Z or provided with this book on a Windows system. Because of the differences in code pages, unpacking can result in corrupting the installation files.

5.2.1 Creating directories on the FTP server and upload the installation image

The compressed file contains the z/VM product DVD. The content of this file must be copied to the directory of the FTP server. Complete the following steps:

1. Start a Secure Shell (SSH) session to the FTP server to be used.
2. Create a directory tree where the files will be stored. We debated over the correct path to use on an FTP server to store the data. The path that you choose to use (`/var/ftp`, `/srv/ftp`, or even `/ftp`) is up to you. You can adjust the FTP server configuration to present any of these paths as `/ftp` for a user. In the example environment that we used to author this book, we opted for simplicity and used `/ftp/zvm/720/`:

   ```
   # mkdir -p /ftp/zvm/720
   ```

3. Set the group ownership of this directory to ***ftp*** or whatever the correct group is on your server to permit the FTP daemon to read the contents. Also, set **+s** for the group so that new files inherit group ownership. In the environment that was used to author this book, the FTP daemon runs under the ID `ftp`, which has a primary group of `ftp`:

   ```
   # chgrp -r ftp /ftp/zvm
   # chmod -r g+s /ftp/zvm
   ```

4. Upload the z/VM installation image from an intermediate workstation, or download it directly from the Internet. The following example shows the use of **rsync** with file attribute retention and transport compression that is enabled to copy the compressed file from an intermediate Linux workstation to an FTP server at the ***<ipaddress>*** to the correct directory. Windows users might want to use an open source FTP application, such as FileZilla or WS-FTP to perform this step:

   ```
   /tmp/vmisodisk/
   bash $ ===> rsync -az --progress *.zip ftproot@<ipaddress>:/ftp/zvm/720
   ...
   ```

   > **Note:** Open source software does not mean unrestricted use in all cases. It is your responsibility to ensure that you are not violating terms, licensing, or any applicable laws. Furthermore, always check with your IT department or Help Desk before you attempt to install any product on a company asset.

5. List the newly copied file:

   ```
   # cd /ftp/zvm/720
   # ls -l
   ```

6. Decompress the file by using the **unzip** command, which creates the subdirectory `cpdvd/`:

   ```
   # unzip CD749500.zip
   ```

   The z/VM product installation `.iso` file is now ready.

Note: In the past, z/VM came with two compressed files. The first file contained the GA level of z/VM and the second file contained the first *Recommended Service Upgrade* (RSU). z/VM V7R2 is only one compressed file that contains the GA installation code and RSU together. Further, the RSU is now applied automatically during installation.

After z/VM is installed, it is strongly recommended that you check the RSU level of the installed system and compare it to the latest available RSU as extra service (updates and fixes) might need to be applied.

7. Create directories for the files that are associated with this book. For more information about these files, see Appendix C, “Additional material” on page 489.
8. Create a directory where the files are to be stored. In our example, we use the following command:

```
# mkdir -p /ftp/zvm/cookbook
```

Note: Previously, this book used an example value of `/ftp/zvm/sg248147/` for this directory. For easy of use, the new value for the directory was adopted. If you created `/ftp/zvm/sg248147/`, use the following commands to move it and then create a symbolic link for it:

```
# cd /ftp/zvm
# mv ./sg248147 ./cookbook && ln -s ./cookbook ./sg248147
```

9. Obtain the latest version of the associated files. You can choose to download the compressed tar archive (.tgz) to an intermediate system such as a workstation, or download it directly from the Internet. The following example shows downloading a compressed tarball that is named ***example*.tgz** to the correct directory directly from the Internet:

```
# cd /ftp/zvm/cookbook
# wget -v www.vm.ibm.com/pubs/redbooks/sg248147/files/example.tgz
...
```

Note: For more information about verifying that you are downloading the latest version of the associated files, see this web page.

10. Expand the tarball file with the **tar** command and the flags for decompression, expansion, and verbosity -- **-zxv**, which creates the subdirectories `lnxadmin`, `lnxmaint` and `maintvrm`:

```
# tar -zxvf example.tgz
...
```

All of the files and utilities that are used throughout this book are ready to be downloaded to your z/VM system, as described in 6.7, “Enabling and configuring RACF”.

5.3 Installing z/VM from a DVD or an FTP server

In the z/VM 7.2.0 Installation Guide, GC24-6292, the following installation types are available to install z/VM:

- Traditional
- Upgrade

The traditional method installs a new z/VM system or VMSSI cluster on a set of DASD, which can then be customized according to your needs.

The upgrade method is used to upgrade from z/VM 6.4 or 7.1 to z/VM 7.2. A new release system is used as a staging system installed in a second-level system of your current system. You must clone your current system to run as a second level, upgrade it, and when complete, the new level of code is moved to the current system with minimal impact. If your current system is an SSI-cluster, you can upgrade one member at a time.

Before installing the new z/VM 7.2, check for current information and make sure that all requirements are satisfied.

> **Tip:** For more information, see the following topics at this web page:
>
> - Important z/VM Installation News
> - z/VM Installation Tips
> - Preventive Service Planning (PSP) bucket for z/VM 720 installation.

If you are not familiar with the Hardware Management Console (HMC) and z/VM, you might want to use the official z/VM manual, *z/VM 7.2 Installation Guide*, GC24-6292, which is available at this web page.

If you are planning a non-SSI installation, see Chapter 7, "Install a z/VM non-SSI LPAR" of the previous version of this publication, *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

If you are installing z/VM second level (z/VM under z/VM) or onto a Fibre Channel Protocol (FCP)/SCSI disk, use the *z/VM 7.2 Installation Guide*, GC24-6292, instead of this Virtualization Cookbook because we do not address those options.

In this book, we describe the load in memory of z/VM. Then, you can decide which of the following types of installations is better for you:

- First-level VMSSI installation of z/VM from DVD or FTP server onto 3390 DASD (highly recommended at least with one VMSSI member only)
- First-level Non-SSI installation of z/VM from DVD or FTP server onto 3390 DASD

5.4 Starting the z/VM installation

An example of the main menu in tree view mode is shown in Figure 5-7. To change between the two HMC views, select **Tasks Index** on the left; then, select **User Settings** on the right, and then, select **UI Style**.

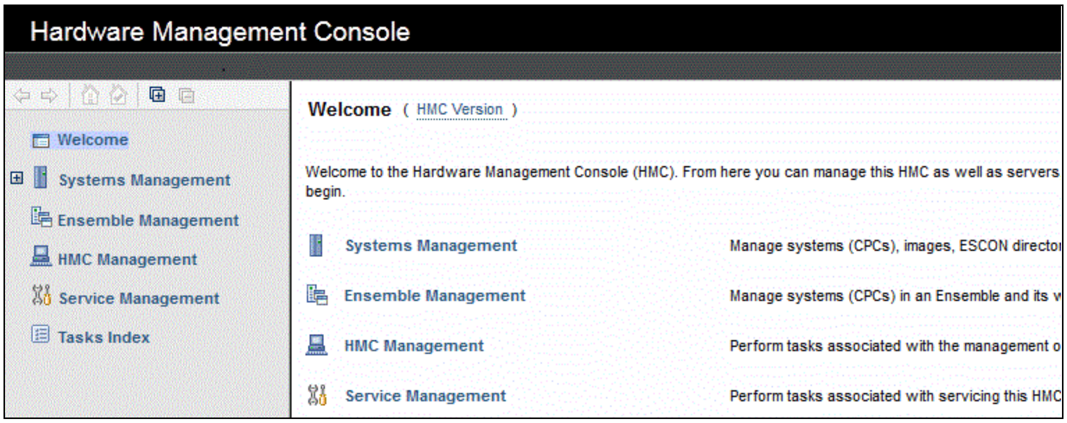


Figure 5-7 HMC tree view

Important: While you work on the HMC, whenever you are prompted for input, such as an IP address, password, parameters, or other value, be aware that pressing **Enter** is the same as clicking **Cancel**. You must click **OK** to submit input.

This safety precaution prevents accidentally committing changes that are destructive. If you press Enter by mistake, repeat any steps that you accidentally canceled.

5.4.1 Logging on to HMC

To begin the z/VM installation, complete the following steps:

1. Log on to the HMC. You need physical access to the console or a URL for the web interface. You also need a user ID and password. Assuming that the view is tree mode, you see a window that is similar to the window that is shown in Figure 5-7.
2. Expand **Systems Management** on the left navigation window. Then, expand **Systems** to view the central processor complexes (CPCs) that are managed by this HMC.
3. Move to the main window on the right side of your window where the LPARs on which you install the VMSSI are shown. Select the LPAR that is to be the *first* member of the z/VM 7.2 VMSSI cluster. The first LPAR (onto which the VMSSI was installed in this example) is shown in Figure 5-8 on page 112. The radio button to the left of the LPAR is selected. In older versions of the HMC, this option might be a check box instead of a radio button.

   Important: The `LOAD` process is *destructive* and cannot be undone. Therefore, you must be certain that you selected the correct LPAR. If you are unsure, check with someone who knows. If you select the wrong LPAR, you irreparably destroy the system that is running on that LPAR.

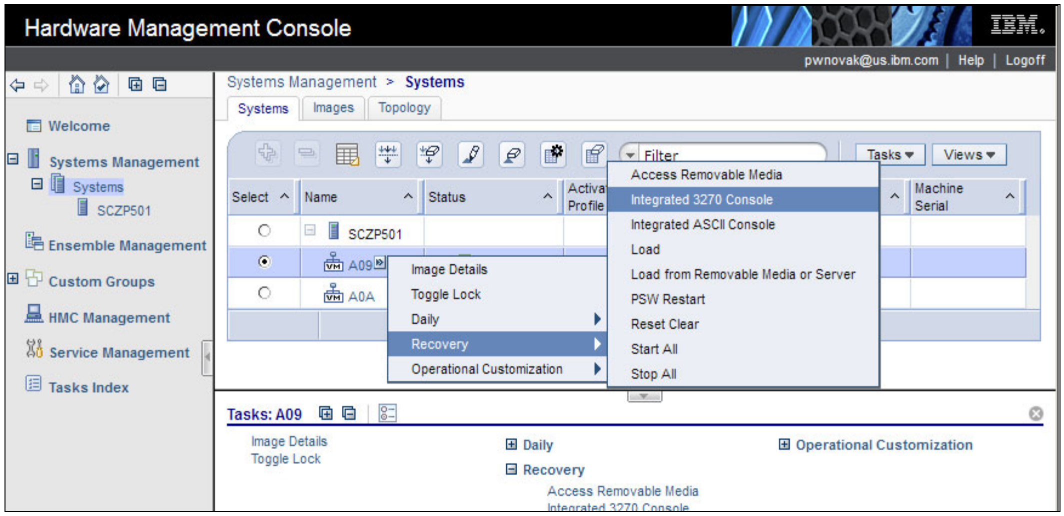


Figure 5-8 HMC with systems selection expanded and Integrated 3270 Console from menu

4. Open an Integrated 3270 Console (see Figure 5-8) by clicking the **Tasks** drop-down menu in the upper right. Then, select **Recovery** → **Integrated 3270 Console**. A new Java window, Integrated 3270 Console, opens.
5. Begin the installation process by selecting **Load from Removable Media or Server** from the same Recovery submenu.
6. The Load from Removable Media or Server window opens, as shown in Figure 5-9. Complete the following steps:
   a. Click **FTP Source**.
   b. Enter the IP address (or hostname) of your FTP server into the Host computer field.
   c. Enter the FTP User ID and Password into those fields.
   d. Enter the FTP server directory path that contains `720vm.ins` into the File location field. In this example, it is `/ftp/zvm/720/cpdvd`.
   e. Click **OK**.

> **If you are installing from DVDs:** The first disc must be in the HMC DVD drive. Click **Hardware Management Console CD/DVD-ROM** only.

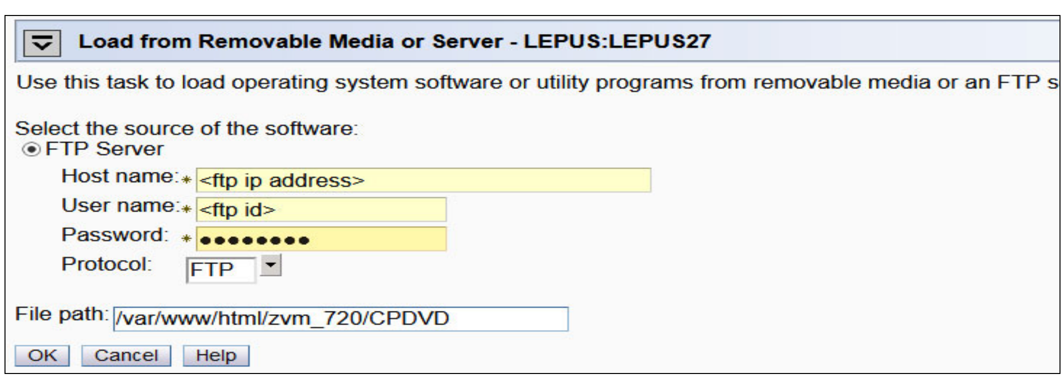


Figure 5-9 HMC FTP Load from Removable Media or Server window

7. Load the RAMDISK:

a. From the Load from Removable Media or Server panel, the directory that contains the file **720VM.INS** is selected. Click **OK**.

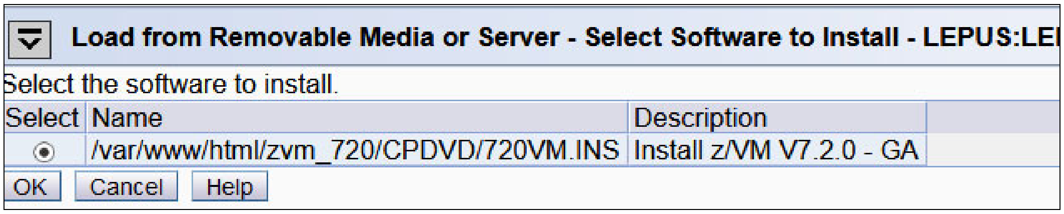


Figure 5-10 HMC FTP load - Select software to install

b. From the Confirm action window, click **OK**

c. From the Continue action window, click **Yes**.

d. You see the Disruptive Task Confirmation: Load from CD-ROM, DVD, or Server Progress window (see Figure 5-11). You might be prompted for the password, depending on your HMC configuration.

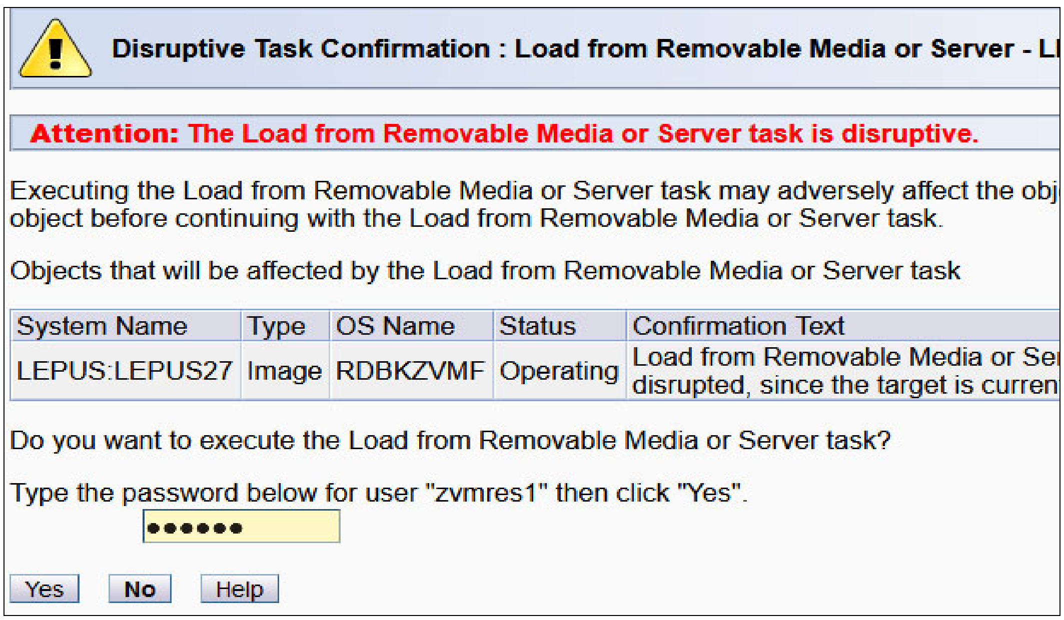


Figure 5-11 HMC FTP load warning

e. You see the Load from Removable media or Server Progress window. When you see the message `Completed successfully`, click **OK** to close. This step takes less than a minute. If you still do not see a message that indicates successful completion after several minutes, repeat Step 7. An in-memory z/VM 7.2 system is running.

5.4.2 In-memory z/VM system loaded

Move to the Integrated 3270 Console window. The RAMdisk IPLs and you see z/VM boot, as shown in Figure 5-12 on page 114. If the Integrated 3270 Console window is still empty, be patient; at times it can take five minutes or more to initialize.

> **Note:** While you are working in the Integrated 3270 Console on the HMC, the Esc key on your keyboard is mapped to the Clear Screen function for the terminal console.

```
12:58:05 z/VM  V7 R2.0  SERVICE LEVEL 0000 (64-BIT)
12:58:06 SYSTEM NUCLEUS CREATED ON 2020-06-26 AT 09:02:09, LOADED FROM $RAMD$
12:58:06
12:58:06 ****************************************************************
12:58:06 * LICENSED MATERIALS - PROPERTY OF IBM*                        *
12:58:06 *                                                              *
12:58:06 * 5741-A09 (C) COPYRIGHT IBM CORP. 1983, 2020. ALL RIGHTS      *
12:58:06 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE,       *
12:58:06 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE     *
12:58:06 * CONTRACT WITH IBM CORP.                                      *
12:58:06 *                                                              *
12:58:06 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES.              *
12:58:06 ****************************************************************
12:58:06
12:58:06 HCPZCO6718I Using parm disk 1 on volume $RAMD$ (device FFFF).
12:58:06 HCPZCO6718I Parm disk resides on blocks 18000 through 52992.
12:58:06 The directory on volume $RAMD$ at address FFFF has been brought
online.
12:58:06 HCPWRS2512I Spooling initialization is complete.
12:58:06 No dump unit - Dump function is SET OFF
12:58:06 HCPAAU2700I System gateway IBMVMRAM identified.
12:58:07 HCPLNM6640E MAINT FFFF not linked.  Minidisk has been defined with the
V mode suffix and is already linked by MAINT.
12:58:07 z/VM Version 7 Release 2.0, Service Level 0000 (64-bit),
12:58:07 built on IBM Virtualization Technology
12:58:07 There is no logmsg data
12:58:07 FILES:   NO RDR,   NO PRT,   NO PUN
12:58:07 LOGON AT 12:58:07 EDT WEDNESDAY 09/22/20
12:58:07 SYSG       LOGON  AS  MAINT    USERS = 1
12:58:07 HCPIOP952I 2G system storage
12:58:07 FILES: 0000001 RDR, 0000001 PRT,      NO PUN
12:58:07 HCPCRC8082I Accounting records are accumulating for userid OPERACCT.
12:58:07 HCPCRC8082I EREP records are accumulating for userid OPEREREP.
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
DMSWSP327I The installation saved segment could not be loaded
z/VM V7.2.0    2020-06-28 13:09
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSVMLIB does not exist
Ready; T=0.01/0.01 12:58:07


                                         RUNNING   IBMVMRAM
```

Figure 5-12   First z/VM 7.2 installation in-memory window

Note: From this point, if you want to install z/VM VMSSI Cluster, see “Installing VMSSI” on page 115. If you want to install a stand-alone system, see “Installing non-SSI z/VM” on page 124.

5.5 Installing VMSSI

The focus of this installation is implementing z/VM VMSSI cluster with 1 - 4 members in 3390 DASDs. We are describing two SSI-members IPL; if you have more members, repeat the instructions of the second member to your third and fourth member.

> **Note:** The information in this section is not to be used for a stand-alone installation or if you want to use VMCSM. For more information about that type of installation, see *5.6, "Installing non-SSI z/VM" on page 124*.

5.5.1 Copying the in-memory z/VM system to DASD

Complete these steps to copy z/VM to DASD:

1. Run the **DVDPRIME** command. The format is **dvdprime *dasdtype* (*source***. The left parenthesis that is shown is part of the command and must be included:
   - For an installation from an FTP server, the ***dasdtype*** is **3390** and the ***source*** is **server**:

     ===> **DVDPRIME 3390 (SERVER**
   - For an installation from a DVD, the ***dasdtype*** is **3390** and the ***source*** is **DVD**:

     ===> **DVDPRIME 3390 (DVD**

   The command completes quickly and you see the following message:

```
DVDPRIME 3390 (SERVER
IUGDVP8327I ** Now executing DVDPRIME on 26 Sep 2020 at 14:19:37 **
IUGDVP8440I Now loading 4CC disk
DVDLOAD: LOADING FILE 'FBA22200 IMAGE *'
DVDLOAD: RC=0
MDREST: WROTE 1800 BLOCKS ON 04CC, RC=0
IUGDVP8392I DVDPRIME EXEC ended successfully
```

   Figure 5-13 DVDPRIME execution messages

2. Run the **INSTPLAN TRADITIONAL** command to set up the configuration for the installation process.

   ===> **INSTPLAN TRADITIONAL**

   If the status at the bottom of the window changes to `HOLDING IBMVMRAM`, you must press the **ESC** to clear the hold.

   You see the z/VM INSTALLATION PLANNING window as shown in Figure 5-14 on page 116.

```
                        *** z/VM INSTALLATION PLANNING ***

 Mark the product(s) selected to be installed into the file pool with an "F"
 and those selected to be installed to minidisks with an "M"
     F        VM             F        DIRM           F        ICKDSF
     F        PERFTK         F        RACF           F        RSCS
     F        TCPIP          F        VMHCD

 Select a System Default Language.
     _ AMENG        _ UCENG

 Select a System DASD type. DASD size can be changed.
     _ 3390  10016     _ FBA DASD  6.0

 Enter the name of common service file pool.
     Filepool Name:     ________

 Select a System Type: Non-SSI or SSI
     _ Non-SSI Install:    System Name ________
     _ SSI Install:        Number of Members _     SSI Cluster Name________



       F1= HELP    F3/F12= QUIT   F5= Process  ENTER= Refresh
```

Figure 5-14 The instplan window

3. It is the recommendation of IBM z/VM Development and the authors of this book that you leave all values set to their default of "**F**" in the top section of this window, as shown in Figure 5-15. Installing to file pools provides you with far more flexibility. Installing to minidisk is a practice that continues to dwindle and is likely to fade further into obscurity.

```
 *** z/VM INSTALLATION PLANNING ***
Mark the product(s) selected to be installed into the filepool with an "F"
and those selected to be installed to minidisks with an "M"
F VM           F DIRM         F ICKDSF
F PERFTK       F RACF         F RSCS
F TCPIP        F VMHCD
Select a System Default Language.
x AMENG     _UCENG
Select a System DASD type. DASD size can be changed.
x 3390  10016 _FBA DASD  6.0
Enter the name of common service filepool.
Filepool Name:    VMPSFS__
Select a System Type: Non-SSI or SSI
_Non-SSI Install:    System Name
xSSI Install:        Number of Members 2    SSI Cluster Name RDBKSSI3
```

Figure 5-15 z/VM INSTALLATION PLANNING

4. Enter the letter `X` next to both AMENG and 3390 Mod 9 (or the type of DASD you use). The common product service file pool default name is `VMPSFS` and it is recommended that you use this value.
5. Leave the `Non-SSI Install` and `System Name` fields in blank. Enter the letter `X` next to `SSI Install`, set the number of members (`2` in this example), and enter a name for the VMSSI Cluster Name (`RDBKSSI3` in this example).
6. You see the VMSSI Cluster Installation window. Read the entire list licensing terms to ensure that you understand them. Then, press **F5** to accept them.
7. The z/VM INSTALLATION PLANNING PANEL 2 appears, as shown in Figure 5-16. Answer No to the question: "`Would you like to have your system automatically configured to be managed by the Unified Resource Manager or some other SMAPI client for system management, such as XCAT or IBM Director?`" by entering **N**. Press **F5** to continue.

IMPORTANT: You must ensure you respond with `N` (no) to the question regarding SMAPI system management. If you respond otherwise, you create a system that you cannot manage by using the concepts and processes that are described in this book.

```
                    *** z/VM INSTALLATION PLANNING PANEL 2 ***

  N_ Would you like to have your system automatically configured to be
      managed by the Unified Resource Manager or some other SMAPI client
      for system management, such as XCAT or IBM Director? (Y/N)

      Keep The Following in Mind:

          If you say YES, you should not attempt to manage your system in
          any other way.

          If you'd like to manage your own system, or use a purchased
          external security manager or a purchased directory manager say NO
```

Figure 5-16 z/VM INSTALLATION PLANNING PANEL 2

8. You see the z/VM INSTALLATION PLANNING PANEL 3, as shown in Figure 5-17. Enter the VMSSI member names and their corresponding LPAR names as seen on the HMC. Press **F5** to continue.

```
                 *** z/VM INSTALLATION PLANNING PANEL 3 ***

  SSI Cluster Name: RDBKSSI03

  After installation is complete, the SSI cluster will be IPLed:

  X    First-Level
  _    Second-Level

  SSI Member Name(s):

  SLOT #     MEMBER NAME    IPL LPAR/USERID
  ======     ===========    ===============
    1         RDBKZVMG        A09
    2         RDBKZVMH        A0A
```

Figure 5-17 z/VM INSTALLATION PLANNING PANEL 3

9. You see a summary of your choices and are prompted whether you want to continue. Carefully review all of the values. If the values are correct, enter `Y` to the question and press **Enter**.

10. You now see the z/VM INSTALLATION VOLUME DEFINITION panel. Initially, it is populated with the default z/VM volume labels `VMCOM1, 720RL1, M0...` as shown in Figure 5-18. Update these default values with the correct information from the planning worksheet that you completed.

```
                 *** z/VM INSTALLATION VOLUME DEFINITION ***

       TYPE      LABEL      ADDRESS                        FORMAT (Y/N)
       COMMON    VMCOM1      ____                                _

       RELVOL    720RL1      ____

       TYPE      LABEL      ADDRESS         TYPE      LABEL      ADDRESS
 RDBKZVMG                                RDBKZVMH
       RES       M01RES      ____           RES       M02RES      ____
       SPOOL     M01S01      ____           SPOOL     M02S01      ____
       PAGE      M01P01      ____           PAGE      M02P01      ____
       WORK      M01W01      ____           WORK      M02W01      ____
```

Figure 5-18 z/VM INSTALLATION VOLUME DEFINITION panel with default labels

Figure 5-19 on page 119 shows the results of the use of the example values from the environment that was used to produce this book. In this example, a prefix character of `V` is used. After you update the label values, press **F5** to continue.

```
                  *** z/VM INSTALLATION VOLUME DEFINITION ***

         TYPE      LABEL      ADDRESS                       FORMAT (Y/N)
         COMMON    RS3CM1 953E Y

         RELVOL    RS3RL1 95BE

         TYPE      LABEL      ADDRESS          TYPE      LABEL      ADDRESS
  RDBKZVMG                                   RDBKZVMH
         RES       RDGRES 963E RES          RDHRES 97BE
         SPOOL     RDGS01 96BE SPOOL        RDHS01 983E
         PAGE      RDGP01 973E PAGE         RDHP01 98BE
         WORK      RDHU01 993E WORK         RDHU01 99BE
```

Figure 5-19 z/VM INSTALLATION VOLUME DEFINITION panel with planning worksheet values

11. You see the z/VM INSTALLATION FIRST-LEVEL CONFIGURATION panel, as shown in Figure 5-20. The common volume addresses almost always are identical. Enter the common volume address for all members and the channel-to-channel (CTC) device addresses.

```
              *** z/VM INSTALLATION FIRST-LEVEL CONFIGURATION ***
 Real addresses for the common volume on each member LPAR:

    VOLUME    DASD       RDBKZVMG    RDBKZVMH
     TYPE     LABEL      ADDRESS     ADDRESS
    =======  ========   =========   =========
     COMMON   RS3CM1 953E 953E

  CTC device addresses:

    From: RDBKZVMG                      From: RDBKZVMH
          To: RDBKZVMG    N/A                 To: RDBKZVMG 1444
          To: RDBKZVMH 1E24 To: RDBKZVMH    N/A
```

Figure 5-20 z/VM INSTALLATION FIRST-LEVEL CONFIGURATION panel

12. Press **F5**. You see a summary of your values. Then the following message is displayed:

```
...HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY.
```

13. Reference your planning worksheet and attach all DASD that is part of the VMSSI cluster to your virtual machine by using the **ATTACH** command. The ***** that is used in the **ATTACH** command means “self” (the ID that is running the command). In this example, we used the following command:

```
===> attach 953E 95BE 963E 96BE 973E 97BE 983E 98BE to *
17:53:14 DASD 953E ATTACHED TO MAINT 953E WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 95BE ATTACHED TO MAINT 95BE WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 963E ATTACHED TO MAINT 963E WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 96BE ATTACHED TO MAINT 96BE WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 973E ATTACHED TO MAINT 973E WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 97BE ATTACHED TO MAINT 97BE WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 983E ATTACHED TO MAINT 983E WITH DEVCTL HYPERPAV BASE
17:53:14 DASD 98BE ATTACHED TO MAINT 98BE WITH DEVCTL HYPERPAV BASE
```

> **Important:** The devices ***953E 95BE 963E 96BE 973E 97BE 983E 98BE*** are in bold italics to signify that you must replace the example values with the correct values from your planning worksheet for your site. This convention is used throughout this book.

14. Run the **INSTALL** command. The DASD is formatted and the z/VM system disks are copied. This step often takes more than one hour:

```
===> install
HCPIIS8490I NOW FORMATTING VOLUME 953E (1 OF ##)
...
```

    You see the message HCPMLP8392I INSTALL EXEC ENDED SUCCESSFULLY.

> **Important**: It is imperative that the **INSTALL EXEC** succeeds. If it fails, do not proceed. You must fix the issues and try again.

15. Run the **INSTSCID REMOVE** command to update the SYSTEM CONFIG file:

```
===> instscid remove
...
MSGPFX8392I INSTSCID EXEC ENDED SUCCESSFULLY
```

16. Run the **SHUTDOWN** command. This command shuts down the last VMSSI member that IPLed. You see the system shutting down, which ends in a disabled wait with a **state code of 961**:

```
===> shutdown
...
HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000 00000961
```

    You see the system identifier in the lower right go back to **IBMVMRAM**, which is the in-memory copy of z/VM that was used to begin the installation process.

17. Shut down the in-memory system:

```
===> shutdown system ibmvmram
16:03:37 SYSTEM SHUTDOWN STARTED
```

    The in-memory copy of z/VM is halted on VMSSI member 1. On the HMC, the LPAR status changes from Operating to Not Operating instead.

z/VM 7.2 is now installed.

5.5.2 IPL the first VMSSI member

IPL your initial z/VM VMSSI system from DASD. Your 3270 Integrated Console session is still running. Perform the following steps to IPL:

1. On the HMC, the LPAR of the first VMSSI member must still be selected. Click the **Tasks** drop-down menu in the upper right, then click the **Recovery** menu, then click **Load**.
2. The Load window opens. Complete the following steps:
   a. Set the Load Address to the new system residence volume, which is 983E in this example.
   b. Set the Load Parameter to **SYSG**, which specifies to use the Integrated 3270 Console.
   c. Click **OK**.
3. When you see the Load Task Confirmation window, click **Yes**.

4. After a minute or less, you see a status of `Success` on the Load Progress window. Click **OK**.
5. Move back to the Integrated 3270 Console window. You see the Stand Alone Program Loader panel as shown in Figure 5-21. Press F10 to continue the IPL of your z/VM system. It might take a while for the system to IPL.

```
 STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 2.0


 DEVICE NUMBER    0963E MINIDISK OFFSET:   39         EXTENT:   1

 MODULE NAME:     CPLOAD     LOAD ORIGIN:       1000


 --------------------------------IPL PARAMETERS----------------------------
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=953E cons=SYSG

-----------------------------------COMMENTS--------------------------------



---------------------------------------------------------------------------


 9= FILELIST   10= LOAD   11= TOGGLE EXTENT/OFFSET
```

Figure 5-21 Stand Alone Program Loader

6. At the Start (Warm|Force|COLD|CLEAN) prompt, enter `warm` and then, press **Enter**:

   ===> **warm**

7. At the Change TOD clock prompt, enter `no`:

   ===> **no**

8. The first VMSSI member IPLs cleanly after about a minute. Disconnect from the OPERATOR virtual machine by using the **DISCONNECT** command:

   ===> **disconnect**

9. Log on MAINT720 user ID and check:

   ===> **q cplevel**

```
q cplevel
z/VM Version 7 Release 2.0, service level 2001 (64-bit)
Generated at 07/29/20 16:50:40 EDT
IPL at 09/20/20 20:13:22 EDT
```

Figure 5-22 First SSI-member q cplevel

===> **q cpload**

```
q cpload
Module CPLOAD was loaded from minidisk on volume RDHRES at cylinder 39.
Parm disk number 1 is on volume RS3CM1, cylinders 1 through 120.
Last start was a system restart from SHUTDOWN REIPL.
```

Figure 5-23 First SSI-member q cpload

10. Disconnect from the MAINT720 virtual machine by using the **DISCONNECT** command:

```
===> disconnect
```

11. Issue the first VMSSI member is now running.

5.5.3 IPL for the remaining VMSSI members

In this example of a two-node VMSSI cluster, only one more member exists. If you are creating a four-member VMSSI cluster, you have three more members.

IPL each of the extra members from the HMC by completing the following steps:

1. Select the next LPAR, again double-checking to ensure that it is the correct choice.
2. Click **Tasks** → **Recovery** → **Integrated 3270 Console**, as shown in Figure 5-24.

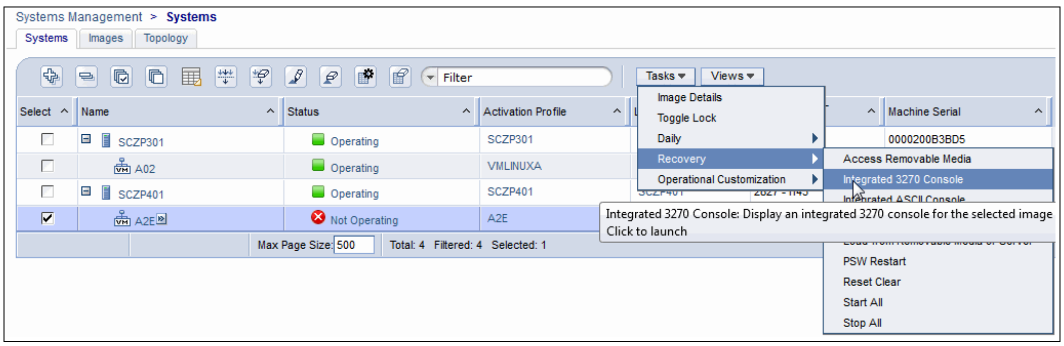


Figure 5-24 Starting a second Integrated 3270 Console

3. Click **Load** in the same Recovery menu. A window opens, as shown in Figure 5-25.

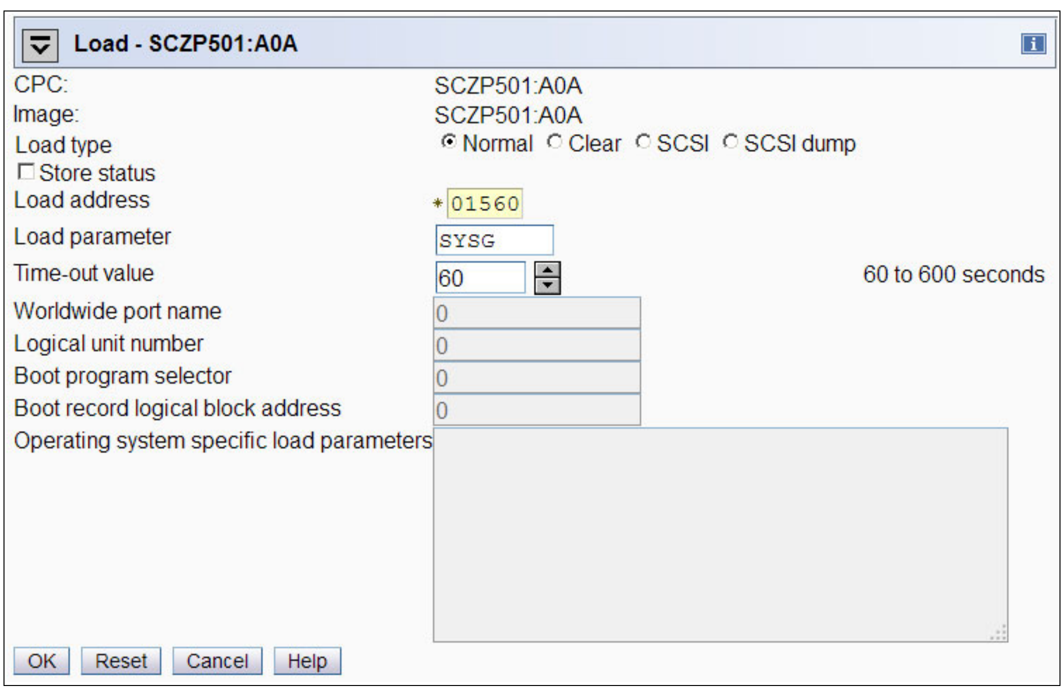


Figure 5-25 Load a second LPAR in the z/VM VMSSI cluster

4. In the Load address field, enter the real device address of the Residence Volume that is allocated to the LPAR on your planning worksheet. In this example, it is `1560`. In the Load Parameter field, enter `SYSG`. Click **OK**.
5. Switch to the Integrated 3270 Console window of the LPAR that you are loading.
6. At the Start (Warm|Force|COLD|CLEAN) prompt, enter `warm`:

   ```
   ===> warm
   ```

7. Because spool data is shared, the warm start typically proceeds without any additional prompts. If you receive a message that states that no warm start data is available, answer the prompt to proceed by using a cold start.
8. After a short time, you see z/VM coming up.

   > **Important:** You might see the following message:
   >
   > `HCPPLM1669I Waiting for ISFC connectivity in order to join the cluster.`
   >
   > This message is *not* acceptable.
   >
   > The member likely waits indefinitely to join. Check with the system administrator and verify that the CTCs are set up correctly and that you used the correct values. Verify that you entered the CTCs correctly. Figure 2-3 on page 62 shows a block diagram of the CTCs that were used in this example.

9. After a minute or two, when z/VM completes the IPL, verify that basic VMSSI awareness is functional by using the **QUERY SSI** command. You see the VMSSI Mode listed as `Stable`, and any z/VM LPARs for which you performed the IPL are in a state of `Joined`. If you do not see these results, troubleshoot your channel-to-channel adapter (CTCA) configuration:

   ```
   ===> query ssi
   SSI Name: ITSOSSIA
   SSI Mode: Stable
   Cross-System Timeouts: Enabled
   SSI Persistent Data Record (PDR) device: VV155A on 155A
   SLOT SYSTEMID STATE     PDR HEARTBEAT       RECEIVED HEARTBEAT
      1 RDBKZVMG Joined    2020-09-19 09:03:01 2020-09-19 09:03:01
      2 RDBKZVMH Joined    2020-00-19 09:03:08 2020-09-19 09:03:08
      3 -------- Available
      4 -------- Available
   ```

10. Run the **DISCONNECT** command to disconnect from the OPERATOR virtual machine on this member; then, close the Integrated 3270 Console window:

    ```
    ===> disconnect
    ```

Repeat these steps until you complete the IPL on all LPARs in your cluster. After the IPLs are complete, all of the members of the VMSSI cluster are up and running.

5.6 Installing non-SSI z/VM

The recommendation is to install one z/VM as a unique member of SSI. If such an installation is your intention, see 5.5, “Installing VMSSI” on page 115.

This installation that is described hers is for a Non-SSI z/VM installation (a stand-alone system).

Before beginning this process, ensure that you completed the steps to load z/VM in memory as described in 5.4, “Starting the z/VM installation” on page 111 with the z/VM in memory loaded. Then, continue with this section.

5.6.1 Copying in-memory z/VM system to DASD

Complete the following steps to copy z/VM to DASD:

1. Run the **DVDPRIME** command. The format is **dvdprime *dasdtype* (*source***:
   - For an installation from an FTP server, the ***dasdtype*** is **3390** and the ***source*** is **server**:

     ===> **dvdprime 3390 (server**
   - For an installation from a DVD, the ***dasdtype*** is **3390** and the ***source*** is **DVD**:

     ===> **dvdprime 3390 (DVD**
2. The command completes quickly and you see the message that is shown in Figure 5-26.

```
dvdprime 3390 (server
IUGDVP8327I ** Now executing DVDPRIME on 26 Sep 2020 at 14:19:37 **
IUGDVP8440I Now loading 4CC disk
DVDLOAD: LOADING FILE 'FBA22200 IMAGE *'
DVDLOAD: RC=0
MDREST: WROTE 1800 BLOCKS ON 04CC, RC=0
IUGDVP8392I DVDPRIME EXEC ended successfully
```

Figure 5-26 DVDPRIME execution messages

3. Run the **INSTPLAN TRADITIONAL** command to set up the configuration for the installation process. You see the z/VM INSTALLATION PLANNING panel, as shown in Figure 5-27. In specific instances, you might need to clear the terminal panel by pressing **Esc** before the INSTPLAN panel appears:

   ```
   ===> instplan traditional
   ```

   It is recommended that you leave the Ms that are in the top section as is for a minidisk installation.

```
                      *** z/VM INSTALLATION PLANNING ***
Mark the product(s) selected to be installed into the filepool with an "F"
 and those selected to be installed to minidisks with an "M"
     F        VM             F        DIRM           F        ICKDSF
     F        PERFTK         F        RACF           F        RSCS
     F       TCPIP           F       VMHCD

 Select a System Default Language.
     X AMENG        _ UCENG

 Select a System DASD type. DASD size can be changed.
     X 3390  10016     _ FBA DASD  6.0

 Enter the name of common service filepool.
     Filepool Name:     VMPSFS__

 Select a System Type: Non-SSI or SSI
     X Non-SSI Install:    System Name RDBKZVMF
     _ SSI Install:        Number of Members _     SSI Cluster Name________




                F1= HELP    F3/F12= QUIT   F5= Process  ENTER= Refresh
```

Figure 5-27 z/VM STANDALONE INSTALLATION PLANNING

4. Enter the letter X next to both AMENG (or select your language) and 3390 Mod 9 (or the type of DASD you use), as shown. The common service file pool default name is VMPSFS and it is recommended that you use this value.

5. Enter the letter `X` next to `Non-SSI Install` and enter your System Name inn the `System Name` field. Leave blank the `SSI Install`, `Number of members`, and `Cluster Name` fields. The information that was selected is displayed, as shown in Figure 5-28.

```
IUGIPX8475I Final selections display
The products you selected to load to minidisk are:
NONE
The products you selected to load to SFS are:
VM DIRM ICKDSF PERFTK RACF RSCS TCPIP VMHCD
The system default language selected:
AMENG
The common service filepool name is:
VMPSFS
The install type you selected is:
Non-SSI
The system name is:
RDBKZVMF
The DASD type you selected to load on is:
3390 -  10016 cylinders
The volumes needed to load z/VM are:
COMMON:  VMCOM1
RELEASE: 720RL1
SYSTEM:  M01RES M01S01 M01P01
Do you want to continue ? (Y/N)
```

Figure 5-28 Final products selection

6. You see a summary of your choices and are prompted whether you want to continue. Carefully review all of the values. If the values are correct, enter `Y` to answer the question and then, press **Enter**.
7. Enter the DASD volumes and addresses in the z/VM INSTALLATION PLANNING PANEL 2, as shown in Figure 5-29.

```
                  *** z/VM INSTALLATION VOLUME DEFINITION ***2

       TYPE      LABEL      ADDRESS                       FORMAT (Y/N)
       COMMON    VMFOM1      923E____                            _
       RELVOL    VMFRL1      92BE____

       TYPE      LABEL      ADDRESS          TYPE      LABEL      ADDRESS
RDBKZVMF
       RES       VMFRES      933E____
       SPOOL     VMFS01      93BE____
       PAGE      VMFP01      943E____
```

Figure 5-29 z/VM Installation DASD volume labels and addresses.

The processing results of this panel are the message that INSTPLAN EXEC ended successfully.

8. Press **F5**. You see a summary of your values, and the following message:

```
...
HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY.
```

9. Reference your planning worksheet and attach all DASDs that are part of the VMSSI cluster to your virtual machine by using the **ATTACH** command. The * that is used in the **ATTACH** command means “self” (the ID that is running the command). In this example, we used the following command:

```
===> attach 923e 92be 933e 93be 943e to *
15:40:57 DASD 923E ATTACHED TO MAINT 923E WITH DEVCTL HYPERPAV BASE
15:40:57 DASD 92BE ATTACHED TO MAINT 92BE WITH DEVCTL HYPERPAV BASE
15:40:57 DASD 933E ATTACHED TO MAINT 933E WITH DEVCTL HYPERPAV BASE
15:40:57 DASD 93BE ATTACHED TO MAINT 93BE WITH DEVCTL HYPERPAV BASE
15:40:57 DASD 943E ATTACHED TO MAINT 943E WITH DEVCTL HYPERPAV BASE
```

> **Important:** The devices that are listed are in bold italics to indicate that you must replace the example values with the values from your planning worksheet for your site. This convention is used throughout this book.

10. Run the **INSTALL** command. The DASD is formatted and the z/VM system disks are copied, as shown in Figure 5-3. This step usually takes more than one hour:

```
===> install
```

Example 5-3 Install process

```
HCPIIS8490I NOW FORMATTING VOLUME 923E (1 OF ##)
...
IUGIIS8380I Restoring IIS to VMFOM1, VMFRL1, VMFRES, and VMFS01
...
IUGILB8440I Now loading PMAINT   2CC (2CC) disk 1 of 192
...
HCPMLP8392I INSTALL EXEC ENDED SUCCESSFULLY.
```

11. You see the message: `HCPMLP8392I INSTALL EXEC ENDED SUCCESSFULLY.`

> **Important**: It is imperative that the **INSTALL EXEC** succeeds. If it fails, do not proceed. You must fix the issues and try again.

12. Run the **SHUTDOWN** command. This command shuts down the last VMSSI member that IPLed. You see the system shutting down, which ends in a disabled wait with a state code of 961:

```
===> shutdown
...
HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000 00000961
```

You see the system identifier in the lower right return to **IBMVMRAM**, which is the in-memory copy of z/VM that was used to begin the installation process.

13. Shut down the in-memory system:

```
===> shutdown system ibmvmram
16:03:37 SYSTEM SHUTDOWN STARTED
```

The in-memory copy of z/VM is halted on VMSSI member 1. On the HMC, the LPAR status changes from `Operating` to `Not Operating` instead.

z/VM 7.2 is now installed.

5.6.2 IPL the new z/VM 7.2

IPL your initial z/VM 7.2 system from DASD. Your 3270 Integrated Console session is still running. Perform the following steps to IPL:

1. In the HMC, the LPAR of the first VMSSI member must still be selected. Click **Tasks** in the upper right; then, click **Recovery** → **Load**.
2. The Load window opens. Complete the following steps:
   a. Set the Load Address to the new system residence volume, which is ***155C*** in this example.
   b. Set the Load Parameter to **SYSG**, which specifies to use the Integrated 3270 Console.
   c. Click **OK**.
3. When you see the Load Task Confirmation window, click **Yes**.
4. After a minute or less, you see a status of `Success` on the Load Progress window. Click **OK**.
5. Move back to the Integrated 3270 Console window. You see the Stand Alone Program Loader panel as shown in Figure 5-21 on page 121. Press **F10** to continue the IPL of your z/VM system. It might take a while for the system to IPL.

5.7 Initial TCP/IP configuration

It is recommended that you initially configure TCP/IP by using the **IPWIZARD** command on each of the VMSSI members. This wizard is generally used only once. After **IPWIZARD** creates the initial configuration files, the files often are maintained manually. A temporary OSA triplet is used to get z/VM in the network. Later, the TCP/IP stack is correctly attached to a highly available z/VM Virtual Switch (VSWITCH).

> **Note:** The TCPIP configuration steps are valid for VMSSI z/VM and Non-SSI z/VM systems.

5.7.1 Using the z/VM IPWIZARD tool

With the `IPWIZARD` tool, you can quickly get z/VM onto an Internet Protocol network.

The **IPWIZARD** command is on the `MAINT 193` disk. You access it on file mode `G` by using the **ACCESS** command so that you pick up **IPWIZARD** from that minidisk. Complete the following steps:

1. Access the `MAINT 193` disk:

   ```
   ===> access 193 G
   ```

2. Invoke **IPWIZARD**:

   ```
   ===> ipwizard
   ```

3. The z/VM TCP/IP Configuration Wizard opens, as shown in Figure 5-30 on page 129:
   a. The first field, User ID, must always be **TCPIP**.
   b. Obtain the remaining values for your installation from Table B-8 on page 486. Our values are shown in Table 2-9 on page 66. Continue by pressing F8.

```
 *** z/VM TCP/IP Configuration Wizard ***
The items that follow describe your z/VM host
User ID of VM TCP/IP Stack Virtual Machine:  TCPIP___
Host Name:    RDBKZVMF____________
Domain Name:  cpolab.ibm.com__________________________
Gateway IP Address: 9.76.61.1_____________________________
DNS Addresses:
1)9.0.128.50___________________________
2)9.0.130.50__
```

Figure 5-30   z/VM TCP/IP Configuration Wizard (panel 1 of 3)

4. Complete the General Interface Configuration Panel that is shown in Figure 5-31 with the following information:
   - Set the Interface Name to **QDIOETH0**, which is recommended, as shown.
   - The Device Number, which is the starting address of the OSA triplet that the z/VM TCP/IP stack uses.
   - The IP address that must be routed to the OSA card, which becomes the IP address of the z/VM system.
   - If you are behind a firewall or other similar type of device that does not permit `ICMP Unreachable` messages (type 3), do not enable the use of Path MTU Discovery (PMTUD). In all other cases, it must be enabled because PMTUD allows the automatic discovery of the correct maximum transmission unit (MTU) during routing for maximum transmission throughput. Check with your network administrator if you are unsure.
   - The Interface Type QDIO (layer 2) with modern OSA devices.

   When you finish, press **F8**.

```
 *** General Interface Configuration Panel ***
Interface Name: OSA1____________ Device Number: 1944
IP Address:     9.76.61.242____
Subnet Mask:    255.255.255.0__
Path MTU Discovery (Optional): _Enabled   _Disabled
Interface Type (Select one):
_QDIO (layer 3)   xQDIO (layer 2)   _LCS
_HiperSockets     _CTC
```

Figure 5-31   General Interface Configuration Panel (configuration wizard panel 2 of 3)

5. In the QDIO Interface Configuration Panel that is shown in Figure 5-32 on page 130, enter the following information:
   - A virtual LAN (VLAN) ID if your network administrator indicated that a VLAN ID is required.
   - The MTU size for an OSA, which is `1492` or `8992` (in this example, `8992`):
     - If your network can support jumbo frames, the use of `8992` provides better performance because OSAs are optimized for this MTU.
     - If you cannot use the PMTUD feature, the use of `1492` is recommended unless you know that your network can use jumbo frames.
   - In general, a value for the Port Number is no longer necessary.

   Press **F5** to complete the wizard.

```
                    *** QDIO Interface Configuration Panel ***

  VLAN ID (optional):  ____


  Maximum Transmission Unit (MTU) size:  1500_


  Port Number (optional):  __
```

Figure 5-32 QDIO Interface Configuration Panel (configuration wizard panel 3 of 3)

The following message displays:

```
DTCIPW2508I DTCIPWIZ EXEC is attempting to create the necessary config. files.
```

6. Enter `1` to restart the TCP/IP stack. (You might also see other warnings.) Watch for the message `HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY`:

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 10 FORCED BY MAINT
...
DTCIPW2519I Configuration complete; connectivity has been verified
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCPIP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

7. Your z/VM TCP/IP stack is up. Ping it from another system. If the **IPWIZARD** fails, you must continue debugging it until it succeeds. Double-check all values. Verify that the Internet Protocol network and OSA information that you were provided are correctly associated.

5.8 Adding CTCAs to an SSI cluster

During installation, the VMSSI CTC installation panel allows two CTC connections to be installed for each SSI member. You must add CTCs for performance and redundancy.

It is recommended that you use four of the eight CTC devices to connect SSI members by way of each channel path. Generally, eight devices are available in a Fibre Channel connection (FICON) CTC control unit. It is recommended that only four of the eight devices are used for performance reasons.

The following example adds three CTCs for each member to each path that was activated during the installation:

- Display the installed CTCs on the first member (`ITSOZVM1`):

```
===> q ctc active
CTCA 47E0 ATTACHED TO SYSTEM  -ISFC
CTCA 57E0 ATTACHED TO SYSTEM  -ISFC
```

- Display the installed CTCs on ITSOZVM2:

```
===> q ctc active
CTCA 4120 ATTACHED TO SYSTEM  -ISFC
CTCA 5120 ATTACHED TO SYSTEM  -ISFC
```

The previous two commands show the four CTCs that were set up during z/VM installation. From these real device addresses, determine the channel paths that they are on by using the following commands:

- Display the channel paths that are used by the CTCs on ITSOZVM1:

```
===> q path to 47e0
Device 47E0, Status ONLINE
CHPIDs to Device 4120 (PIM)  : 4C
===> q path to 57e0
Device 57E0, Status ONLINE
CHPIDs to Device 57E0 (PIM)  : 4D
```

- Display the channel paths that are used by the CTCs on ITSOZVM2:

```
===> q path to 4120
Device 4120, Status ONLINE
CHPIDs to Device 4120 (PIM)  : 4C
===> q path to 5120
Device 5120, Status ONLINE
CHPIDs to Device 5120 (PIM)  : 4D
```

The previous two commands show the channel-path identifiers (CHPIDs) that the CTCs are on. In this example, they are 4C and 4D. From these CHPIDs, determine the other available CTC devices by completing the following steps:

1. Display the devices that are used by the channel paths on ITSOZVM1:

```
===> q chpid 4c
Path 4C online to devices 47E0 47E1 47E2 47E3 4A90 4A91 4A92 4A93
===> q chpid 4d
Path 4D online to devices 57E0 57E1 57E2 57E3 5A90 5A91 5A92 5A93
```

2. Display the devices that are used by the channel paths on ITSOZVM2:

```
===> q chpid 4c
Path 4C online to devices 4120 4121 4122 4123 4A90 4A91 4A92 4A93
===> q chpid 4d
Path 4D online to devices 5120 5121 5122 5123 5A90 5A91 5A92 5A93
```

It is recommended to confirm with your hardware configuration engineer that you can add three CTCs to each channel path on each z/VM member. They must be added both dynamically and permanently. Next, run the following commands:

1. Verify that the next three CTCs are available on ITSOZVM1:

```
===> q 47e1 47e2 47e3
CTCA 47E1 FREE    , CTCA 47E2 FREE    , CTCA 47E3 FREE
===> q 57e1 57e2 57e3
CTCA 57E1 FREE    , CTCA 57E2 FREE    , CTCA 57E3 FREE
```

2. Verify that the next three CTCs are available on ITSOZVM2:

```
===> q 4121 4122 4123
CTCA 47E1 FREE    , CTCA 47E2 FREE    , CTCA 47E3 FREE
===> q 5121 5122 5123
CTCA 5121 FREE    , CTCA 5122 FREE    , CTCA 5123 FREE
```

You now have the real device addresses of the CTCs to add to each SSI member.

5.8.1 Adding the CTC devices dynamically

To add the CTC devices dynamically, complete the following steps:

1. Log on to `MAINT` on the first member.
2. Activate six CTCs on the first member, `ITSOZVM1`:

```
===> activate islink 47e1 47e2 47e3 57e1 57e2 57e3
Link device 47E1 activated.
Link device 47E2 activated.
Link device 47E3 activated.
Link device 57E1 activated.
Link device 57E2 activated.
Link device 57E3 activated.
```

3. Activate six CTCs on `ITSOZVM2`:

```
===> activate islink 4121 4122 4123 5121 5122 5123
Link device 4121 activated.
Link device 4122 activated.
Link device 4123 activated.
Link device 5121 activated.
Link device 5122 activated.
Link device 5123 activated.
```

   When the device is active on both systems, you see a `HCPKCL2714I` message. You see the added CTCs if you reissue the **`QUERY CTC`** command.

4. Issue the **`QUERY CTC`** command from `ITSOZVM1`:

```
===> q ctc
CTCA 47E0 ATTACHED TO SYSTEM  -ISFC
CTCA 47E1 ATTACHED TO SYSTEM  -ISFC
CTCA 47E2 ATTACHED TO SYSTEM  -ISFC
CTCA 47E3 ATTACHED TO SYSTEM  -ISFC
CTCA 57E0 ATTACHED TO SYSTEM  -ISFC
CTCA 57E1 ATTACHED TO SYSTEM  -ISFC
CTCA 57E2 ATTACHED TO SYSTEM  -ISFC
CTCA 57E3 ATTACHED TO SYSTEM  -ISFC
```

5. Issue the **`QUERY CTC`** command from `ITSOZVM2`:

```
===> q ctc
CTCA 4120 ATTACHED TO SYSTEM  -ISFC
CTCA 4121 ATTACHED TO SYSTEM  -ISFC
CTCA 4122 ATTACHED TO SYSTEM  -ISFC
CTCA 4123 ATTACHED TO SYSTEM  -ISFC
CTCA 5120 ATTACHED TO SYSTEM  -ISFC
CTCA 5121 ATTACHED TO SYSTEM  -ISFC
CTCA 5122 ATTACHED TO SYSTEM  -ISFC
CTCA 5123 ATTACHED TO SYSTEM  -ISFC
```

This output shows that the CTC devices were added dynamically.

5.8.2 Adding the CTC devices permanently

To add the CTC devices to the SSI permanently, complete the following steps:

1. Log on to `MAINT` on the first SSI member.
2. Access the `PMAINT CF0` disk read/write and link as file mode `F`:

   ```
   ===> link pmaint cf0 cf0 mr
   ===> acc cf0 f
   ```

3. Make a backup copy of the `SYSTEM CONFIG` file:

   ```
   ===> copy system config f = confwrks = (rep
   ```

4. Edit the `SYSTEM CONFIG` file and find the `ISLINK` statements by using the **`/Activate ISLINK`** subcommand. Change the `ISLINK` statements to include the new CTCs. `BEGIN` and `END` statements are added because the new values require two lines each:

   ```
   ===> x system config f
   ====> /activate islink
   ```

The following examples show the `SYSTEM CONFIG` file before and after the changes are made.

The `SYSTEM CONFIG` file looks like the following example *before* the changes are made:

```
/**********************************************************************/
/*         Activate ISLINK statements                                 */
/**********************************************************************/

ITSOZVM1:      ACTIVATE ISLINK 47E0 57E0  NODE ITSOZVM2
ITSOZVM2:      ACTIVATE ISLINK 4120 5120  NODE ITSOZVM1
```

The `SYSTEM CONFIG` file looks like the following example *after* the changes are made:

```
/**********************************************************************/
/*         Activate ISLINK statements                                 */
/**********************************************************************/

ITSOZVM1:      BEGIN
   ACTIVATE ISLINK 47E0 47E1 47E2 47E3  NODE ITSOZVM2
   ACTIVATE ISLINK 57E0 57E1 57E2 57E3  NODE ITSOZVM2
ITSOZVM1:      END
ITSOZVM2:      BEGIN
   ACTIVATE ISLINK 4120 4121 4122 4123  NODE ITSOZVM1
   ACTIVATE ISLINK 5120 5121 5122 5123  NODE ITSOZVM1
ITSOZVM2:      END
```

When the system is restarted, the `ISLINKs` are active between members.

5.8.3 Configuring TCPIP to automatically start during the system IPL

Complete the following steps:

1. Use `VMLINK` to access the `191` disk for `AUTOLOG1` in `FILELIST`:

   ```
   ===> vmlink autolog1 0191 < * * MR > (filelist
   ```

2. FILELIST starts and your cursor is on the line that contains `PROFILE EXEC`. If it is not, move it to that line. Press **PF11** to open the `PROFILE EXEC` in `XEDIT`.

3. Move to the line that begins with Customer processing. Move down one line, and then, insert an `XAUTOLOG` statement for the TCPIP ID:

```
====> /Customer processing
====> down 1
====> input "PIPE CP XAUTOLOG TCPIP"
```

   Your results must look like the following example:

```
/*******************************************************************/
/* Customer processing can be added here                          */
/*******************************************************************/
"PIPE CP XAUTOLOG TCPIP"
```

4. Save your changes and quit XEDIT:

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

5. You are now returned to FILELIST. Press **PF3** to quit FILELIST and return to CMS. VMLINK then automatically releases and detaches AUTOLOG1 191 for you:

```
DMSVML2061I AUTOLOG1 0191 detached
```

6. Run the **`LOGOFF`** command so that the `PMAINT 2CC` disk is freed.

> **Important:** For all other members in the VMSSI cluster, you must now repeat all of section 5.7, "Initial TCP/IP configuration" on page 128. When you run `IPWIZARD` on the other members, you see that the network information is pre-populated with the values from the last node it was run on. Be sure to change the values for each node.

All members of the VMSSI cluster are now accessible by network over TCP/IP.

It is recommended to discontinue the use of the Integrated 3270 Console through the HMC and instead access your new systems with a correct 3270 emulator. For more information, see , "IBM 3270 emulators" on page 471.

To switch to a 3270 emulator, ensure that you issued **`LOGOFF`** from any Integrated 3270 Console sessions that might still be open.

```
 STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 2.0


 DEVICE NUMBER    0933E MINIDISK OFFSET:   39         EXTENT:   1

 MODULE NAME:     CPLOAD     LOAD ORIGIN:       1000


 --------------------------------IPL PARAMETERS-----------------------------
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=923E cons=SYSG

-----------------------------------COMMENTS---------------------------------



----------------------------------------------------------------------------

 9= FILELIST   10= LOAD   11= TOGGLE EXTENT/OFFSET
```

Figure 5-33 Stand Alone Program Loader

7. At the Start (Warm|Force|COLD|CLEAN) prompt, enter `warm` and press **Enter**.

```
===> warm
```

8. At the Change TOD clock prompt, enter `no`:

```
===> no
```

9. The first VMSSI member IPLs cleanly after approximately one minute. Disconnect from the OPERATOR virtual machine by using the **DISCONNECT** command:

```
===> disconnect
```

The new z/VM 7.2 is now running with TCIP configured and you are now ready to configure z/VM.

6

Configuring z/VM

This chapter describes configuring z/VM 7.2 as a two-node VM Single System Image feature (SSI) cluster after the installation is complete. This configuration process includes the initial configuration, hardening, and enabling basic system automation.

If you are new to z/VM and plan to install on only one logical partition (LPAR), it is still recommended that you proceed by using the SSI path. The installation of a single-member SSI cluster provides you with a path for future expansion to add member nodes later.

If you are going to install to SCSI disk, you cannot install with VMSSI enabled. However, non-SSI installations can take advantage of the Centralized Service Management (CSM) capability that was introduced with z/VM 7.2. CSM allows non-SSI installations to manage service across a set of systems from one central system.

In this book, we do not describe the non-SSI installation in detail; however, we do describe setting up VMCSM in Chapter 9, “z/VM Centralized Service Management” on page 275.

Important: Order matters. Prevent unnecessary problems and rework by fully completing all tasks in this volume in the order they are presented. Each chapter relies upon actions that are taken in the previous chapters. Also, s specifically sequenced to give you the quickest results.

This chapter includes the following topics:

6.1 Configuring z/VM

Pay close attention to the following files when configuring z/VM:

- The SYSTEM CONFIG file, which is on the parm disk. This CMS-formatted minidisk can be read by a Control Program (CP). For more information, see 6.3, “z/VM parm disks” on page 143.
- LOGO CONFIG file, which also is on parm disk.
- USER DIRECTORY, which can be on MAINT 2CC or controlled by DIRMAINT.

The following configuration changes that are on parm disk are read only when you IPL z/VM:

- Dynamic system changes (in SYSTEM CONFIG) are made by using Control Program (CP) commands.
- Logo configuration changes (LOGO CONFIG) by using the CP **`refresh`** command.

The z/VM User Directory configures virtual machines or user IDs. It is read often by the system and can be dynamically updated.

In this section, we discuss the SYSTEM CONFIG and USER DIRECTORY files.

6.2 Configuring the XEDIT PROFILE

z/VM uses a program that is called XEDIT as the text editor for the system. It is similar in function to `vi/vim`, `EMACS`, nano, or `pico` on Linux. When XEDIT is started, it looks for the configuration file `XEDIT PROFILE`. Not all CMS virtual machines always have a copy of this file; therefore, XEDIT sessions can look and behave differently, which can be a problem. The steps in this section help resolve this issue for you.

If you are unfamiliar with XEDIT, a cheat sheet is available in Appendix B, “Reference, cheat sheets, blank worksheets, and education” on page 475. This appendix also includes the URL to the z/VM Library where you can obtain more information.

This section guides you in the configuration of the XEDIT profile for system-wide usage. More importantly, these steps also provide the understanding to use XEDIT functions to add, move, and change text. You use XEDIT substantially through the rest of this book and in the administration of your z/VM environment. The efforts that you spend to customize XEDIT result in a much higher level of usability, and make editing easier and faster.

The `191 (A)` disks for MAINT and MAINT720 have a basic version of `PROFILE XEDIT`. When you edit files while you are logged in as either of these user IDs, the values in the profile are usually in effect. Example 6-1 shows how to view this basic profile.

Example 6-1 Original MAINT/MAINTvrm XEDIT profile before it is edited

```
===> type profile xedit
********** THIS IS THE REAL THING ***********
SET NUM ON
SET NULLS ON
SET CASE M I
SET SERIAL OFF
SET PF3 QUIT
SET PF7 BACK
SET PF8 FORWARD
```

```
SET PF9 SPLTJOIN
SET PF10 RIGHT 10
SET PF11 LEFT 10
SET PF12 FILE
SET PF23 SPLTJOIN
SET CMDLINE BOTTOM
SET CURLINE ON 3
SET SCALE OFF
SET STAY ON
```

To configure the default XEDIT profile for use across the entire SSI cluster, complete the following steps:

1. Log on as `MAINT720` on the first SSI member.
2. Back up the existing `PROFILE XEDIT`:

   ```
   ===> copy profile xedit a profile xediorig a (olddate
   ```

3. Update the `PROFILE XEDIT` file:

   ```
   ===> xedit profile xedit
   ```

   a. Change the comment line at the top of the file so that the comment line indicates the name and purpose, and the date and name or ID of the person who last modified it. This task is a preferred practice.

      Make this step a habit for all CMS REXX EXEC files that you edit. Type over the entire first line to replace it with the following information. Replace `YYYY-MM-DD` with today's date, and replace `MYUSERID` with something that is unique to yourself. Be sure to include the `/*` at the beginning and the `*/` at the end:

      /* DEFAULT PROFILE XEDIT FOR z/VM -- MOD *YYYY-MM-DD MYUSERID* ***/**

   b. One default setting that can be dangerous, especially if you use F12 to retrieve commands, is that PF12 is set to the **FILE** (save and quit) subcommand. Most times, you do not want to save your changes and quit with the stroke of one key. It is recommended that you instead set PF12 to the **?** (retrieve) subcommand, which effectively retrieves the last command that was issued on the XEDIT command line. Change the line `SET PF12 FILE` to:

      SET PF12 ?

   c. Press **Enter** to move to the command line (====>) at the bottom of the window.

   d. Because our active XEDIT session was started by using the unchanged profile, we must define PF12 as RETRIEVE for the active session. Enter the subcommands that are shown to set the definition and then, verify the result:

      ```
      ====> set pf12 ?
      ====> query pf12
      ```

      The active definition for PF12 appears at the top of the window, as shown in Figure 6-1.

```
 PROFILE  XEDIT    A1  V 255  Trunc=255 Size=120 Line=0 Col=1 Alt=0
PF12     ONLY    ?

...

====> query pf12
                                                X E D I T  1 File
```

Figure 6-1 Output from XEDIT that displays the active definition for PF12

e. Save the changes up to this point to disk:

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

f. Enter the following subcommands to find the string **SET** and replace it with **'SET** instead. Then, move back to the top of the file. The **CHANGE** command is equivalent to **1,$s/SET/'SET/g** in Linux vi or vim:

```
====> top
====> change/SET/'SET/* *
```

g. Move to line number 2. Then, use the **CAPPEND** (character append) macro to add a closing single quotation mark to the end of the line. The following command is `CAPPEND` followed by a space and then, a single quotation mark:

```
====> :2
====> cappend '
```

h. Use the number sign (**#**) to chain two commands together and append the single quotation mark to the end of the next line. (Note the space and single quotation mark that follow `CAPPEND`):

```
====> down 1 # cappend '
```

Use the repeat function that is assigned to PF12 in an earlier step and repeat this step until each line that begins with **'SET** has a closing single quotation mark.

i. Press **Enter** twice to move back to the command line. Enter the following subcommands to move to the last line in the file and then enable INPUT mode:

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

j. You are now in INPUT mode, where you can enter multiple lines of text. Enter the following lines of text and press **Enter** after each line. Include all special characters and punctuation marks that are shown, such as quotation marks and equal signs:

```
'SET COLOR CURLINE YE REV'
'SET COLOR PREFIX BL NO'
RDK = 'PF1-HELP 3-Quit 7-PgDn 8-PgUp 9-SpJn 10-R10 11-L10 12-Repeat'
'SET RESERVED -2 WH HI 'RDK
```

Tip: If you are reading an electronic copy of this book, you might paste the entire block of text from the book directly into your 3270 XEDIT session while in INPUT mode.

k. Press **Enter** twice to exit out of INPUT mode and return back to the command line. Enter the subcommand **FILE** and press **Enter** again to save your changes, quit XEDIT, and return to CMS and the ready prompt:

```
====> file
Ready;
```

Before you edit your PROFILE XEDIT, it looks similar to the output that is shown in Example 6-1 on page 139. After you edit it, your PROFILE XEDIT looks similar to Example 6-2.

Example 6-2 Modified MAINT/MAINTvrm XEDIT profile with example date and user ID shown

```
===> type profile xedit
/*** DEFAULT PROFILE XEDIT FOR z/VM -- MOD 2015-04-06 PWNOVAK ***/
'SET NUM ON'
'SET NULLS ON'
'SET CASE M I'
```

```
'SET SERIAL OFF'
'SET PF3 QUIT'
'SET PF7 BACK'
'SET PF8 FORWARD'
'SET PF9 SPLTJOIN'
'SET PF10 RIGHT 10'
'SET PF11 LEFT 10'
'SET PF12 ?'
'SET PF23 SPLTJOIN'
'SET CMDLINE BOTTOM'
'SET CURLINE ON 3'
'SET SCALE OFF'
'SET STAY ON'
'SET COLOR CURLINE YE REV'
'SET COLOR PREFIX BL NO'
RDK = 'PF1-HELP 3-Quit 7-PgDn 8-PgUp 9-SpJn 10-R10 11-L10 12-Repeat'
'SET RESERVED -2 WH HI 'RDK
```

4. Complete the following steps to make the modified file available to other virtual machines by copying it to the `MAINT 19E` disk with file mode suffix `2`:
   a. Release the current `19E` disk:

      `===>` **`release 19E`**

   a. Use VMLINK to obtain the `MAINT 19E` disk read/write as file mode **`F`**:

      `===>` **`vmlink maint 19E < 19E F MR >`**

   b. Copy it to the `MAINT 19E` disk (`F`) with file mode suffix 2. (Because the `MAINT 19E` disk is commonly accessed with a file mode suffix of `2`, files are not seen by other virtual machines unless they have this file mode suffix.)

      `===>` **`copy profile xedit A = = F2`**

   c. Save the CMS named saved segment (NSS) with the following commands. Do not be concerned if the numeric value that you see for the `fileid` is different on your system from the example that is shown (different numeric values are normal):

```
===> access 193 G
===> sampnss cms
HCPNSD440I The Named Saved System (NSS) CMS was successfully defined in
field 0017.
===> ipl 190 parm savesys cms
HCPNSS440I Named Saved System (NSS) CMS was successfully saved in fileid
0017.
```

5. LOGOFF as MAINT720 from the current member.
6. Repeat Step 4 on all other members in the SSI cluster.

The same `XEDIT PROFILE` is now accessible to all virtual machines in the SSI cluster.

Note: A copy of PROFILE XEDIT is included in the additional materials that are supplied with this book. This copy contains all of the depicted changes. If you are familiar with XEDIT, you might want to use the contents of that file.

6.3 z/VM parm disks

The system definition information that is required at IPL is in files on the parm disk, a CMS-formatted minidisk that CP can read. The following parm disks are available, and this layout is the same for non-SSI and SSI installations to help ease the migration to a later SSI environment:

- PMAINT CF0 is the parm disk where the main system configuration file and the logo configuration file are located. If you use ECKD DASD, this disk is on the common volume (default label VMCOM1):
  - The main system configuration file, usually called SYSTEM CONFIG, contains operating characteristics, such as the layout of the CP system residence disk, lists of DASD volumes that CP uses, your real storage configuration, and information CP requires to determine the correct offset from Coordinated Universal Time (UTC).

    It also contains real device definitions for I/O devices that do not respond to a sense ID request and for I/O devices that need more information that is defined than a sense ID request returns (for example, printers and communications controllers).
  - The logo configuration file, usually called LOGO CONFIG, contains information about the creation and configuration of logos, including the file names and file types of the different logo files. For more information about setting or modifying this file, see 15.2, “Modifying the z/VM LOGON panel” on page 442.
- MAINT CF1 is the parm disk where the CPLOAD MODULE is stored. This CP kernel file is loaded at IPL. If you use ECKD DASD, this disk is on the system residence volume (default label M01RES).
- MAINT720 CF2 (where 720 is the z/VM version, release, and modification level) is the parm disk that serves as a staging area for updates that are applied by the SERVICE command before you use PUT2PROD to copy them to MAINT CF1. If you use ECKD DASD, this disk is on the release volume (default label vrmRL1).

6.4 System Configuration file

The System Configuration file is one of the most important files of z/VM. It contains operating characteristics, such as the layout of the system residence disk, real storage, and I/O devices configuration and the description of other resources that are available to the system.

The system configuration file is on a partition of a volume that is called *minidisk* and it is allocated as PARM. This minidisk is under user ID PMAINT, and it is address is CF0. The file is called SYSTEM CONFIG by default, although you can change the name in your installation.

Tip: As a best practice, always run a CPSYNTAX check after modifying SYSTEM CONFIG to check whether errors exist on this file.

6.5 Editing the z/VM SYSTEM CONFIG file

The first configuration file that is read when z/VM IPLs is the `SYSTEM CONFIG` file. Only one `SYSTEM CONFIG` file exists for each SSI cluster. As a system programmer, you must become familiar with the `SYSTEM CONFIG` file. The `SYSTEM CONFIG` file contains the primary system definitions that are used when the control program (CP) is booted (IPL). All of the information that is 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) that belongs to the PMAINT user ID. Minidisks that contain such objects are called “parameter (parm) disks” because when they are allocated, those disks are given a special record category type called “PARM”. More than one parm disk can be allocated in a z/VM system for backup and recovery.

More subcommands for XEDIT also are described.

6.5.1 Modifying features and optimizing parameter settings

The FEATURES statement in SYSTEM CONFIG allows you to modify attributes that are associated with the running system at IPL time. Some defaults were changed on z/VM 7.2. The following important features options were changed:

- The Auto_Warm_IPL feature causes CP to bypass prompting for start options if the previous system shutdown was successful. The feature allows for a fully automated startup of z/VM.

  > **Important:** If you are planning to use an External Security Manager (ESM), such as IBM Resource Access Control Facility for z/VM (RACF/VM), you must not enable Auto Warm IPL until your ESM is fully configured.

- 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. This feature prevents another user who might define an identically sized temporary disk from accessing data that was written by the previous user.

  > **Note:** If you are planning to use Temporary Disks (T-Disk), for compliance reasons, modify the default on SYSTEM CONFIG, which enables the clean up.

- The Retrieve feature defines the default and maximum number of retrieve buffers that are allowed per user on your system. Retrieve buffers create a command history, from which users can retrieve commands that were issued. Command retrieval often is assigned to a program function key, such as PF12 (F12).

  > **Note:** 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 likely do not need to change these settings.

- The Passwords_on_Cmds feature tells CP whether to prompt users for passwords when the **`CP AUTOLOG`**, **`LINK`**, or **`LOGON`** commands are used.

  > **Note:** The z/VM 7.2 default for Password_on_Cmds were changed to NO.

- The Disconnect_timeout feature controls whether and when a virtual machine is logged off after it is forced to disconnect. You turn off this feature so that any virtual machine that was forced to disconnect is note logged off.

Note: This feature logs off the user ID that is disconnected after a time out period.

- 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. CP waits to shut itself down until the time interval (in seconds) is exceeded, or all of the virtual machines that are enabled for the signal shutdown reported a successful shutdown. Some Linux distributions support this function, which allows Linux to shut down cleanly before z/VM shuts down.

To modify attributes, complete the following steps as `MAINT720`:

1. Use `VMLINK` to access the `PMAINT CF0` disk as multi-read/write (**`MR`**) and file mode **`F`**. Include the left less than symbol or angle bracket (<) and the right greater than symbol or angle bracket (>) in your command:

```
===> VMLINK PMAINT CF0 < * F MR >
DMSVML2060I PMAINT CF0 linked MR as 0120 file mode F
===> VMLINK MAINT 193 < * G RR >
DMSVML2060I MAINT 193 linked RR as 0121 file mode G
```

2. Review the file information for files that match `SYS* CONF* F`:

```
===> listfile sys* conf* F (ISO
FILENAME FILETYPE FM FORMAT LRECL  RECS  BLOCKS   DATE       TIME
SYSTEM   CONFIG   F1 V      80    378   5        2020-09-19  09:48:05
```

3. Back up the plain `SYSTEM CONFIG` file by using the **`COPYFILE`** command with the **`OLDDATE`** parameter so that the time stamp of the file is not modified. Because the target file name (`SYSTEM`) and mode (`F`) are the same, the equal sign (`=`) can be used to indicate that the value from the source file is reused for the target:

```
===> copy system config f = conforig = (olddate
```

Important: Before any SYSTEM CONFIG change, always make a backup copy on parm disk. This file is read during the IPL and any error can prevent the completion of the z/VM IPL.

4. Check to ensure that your backup is present:

```
===> listfile sys* conf* F (ISO
FILENAME FILETYPE FM FORMAT LRECL  RECS  BLOCKS   DATE       TIME
SYSTEM   CONFIG   F1 V      80    378   5        2020-09-19  09:48:05
SYSTEM   CONFORIG F1 V      80    378   5        2020-09-19  09:48:05
```

5. Open the original file in XEDIT and make the following changes, which are shown in Example 6-6 on page 147:

```
===> xedit system config f
```

   a. Jump to the line that contains the `Features` statement by using the search (forward slash (/)) subcommand, which works like vi / vim works under Linux:

```
====> /Features
```

   The Features section of SYSTEM CONFIG shows the following attributes:

```
Features ,
   Retrieve ,                    /* Retrieve options              */
     Default  20 ,               /* Default.... default is 20     */
     Maximum  255 ,              /* Maximum.... default is 255    */
   MaxUsers noLimit ,            /* No limit on number of users   */
```

```
   Vdisk Userlim 144000 blocks,    /* Maximum vdisk allowed per user */
   Disconnect_Timeout 15,          /* Can be OFF, default is 15 min  */

   Enable ,                        /* Enable the following features  */
     New_Devices_Initialized_When_Added,  /* Make new devices online */

   Disable ,                       /* Disable the following features */
     Set_Privclass ,               /* Disallow SET PRIVCLASS command */
     Auto_Warm_IPL ,               /* Prompt at IPL always           */
     Clear_TDisk   ,               /* Don't clear TDisks at IPL time */
     Validate_Shutdown ,           /* Don't require system name      */
     STP_Timezone ,                /* STP feature is not used        */
     Paging_Alias ,           /* HyperPAV alias not used for paging  */
     Paging_HPF       /* High Performance FICON not used for paging */
```

b. Move the entry to Clear_TDisk from the Disable section to Enable by running the following actions, shown in Example 6-3:

```
====> /Clear_TDisk
====> On prefix area, put M (of Move) in this line
====> On prefix area, put F (of After) in the line you want to move it to
```

Example 6-3 Moving one line inside SYSTEM CONFIG (Features)

```
00150  Features ,
00151    Retrieve ,
00152      Default  20 ,
00153      Maximum  255 ,
00154    MaxUsers noLimit ,
00155    Vdisk Userlim 144000 blocks,
00156    Disconnect_Timeout 15,
00157
00158    Enable ,
f0159      New_Devices_Initialized_When_Added,
00160
00161    Disable ,
00162      Set_Privclass ,
00163      Auto_Warm_IPL ,
m0164      Clear_TDisk   ,
00165      Validate_Shutdown ,
00166      STP_Timezone ,
00167      Paging_Alias ,
00168      Paging_HPF
```

```
====> enter
```

The results of these actions are shown in Example 6-4.

Example 6-4 Results of moved line inside SYSTEM CONFIG file (Features)

```
00150  Features ,
00151    Retrieve ,
00152      Default  20 ,
00153      Maximum  255 ,
00154    MaxUsers noLimit ,
00155    Vdisk Userlim 144000 blocks,
00156    Disconnect_Timeout 15,
```

```

  Enable ,
    New_Devices_Initialized_When_Added,
      Clear_TDisk   ,

  Disable ,
    Set_Privclass ,
    Auto_Warm_IPL ,
    Validate_Shutdown ,
    STP_Timezone ,
    Paging_Alias ,
    Paging_HPF
```

c. Under the `RETRIEVE` stanza, change `Default 20` to the following value:

   `Default 99`

d. Set the **`Disconnect_Timeout`** to **`off`** so that disconnected IDs do not get forced off.

e. Update the amount of time you want to give to z/VM and to Linux servers before a z/VM shutdown is completed. To find the shutdown time, run the following command:

   `====>` **`/ShutdownTime`**

   Update the Shutdowntime amount to 60 by typing over the current value. Update the Signal ShutdownTime to 300 by typing over the current value (see Example 6-5):

Example 6-5 Shutdowntime and Signal Shutdowntime values

```
/*********************************************************************/
/*                    Set Shutdown time periods                      */
/*********************************************************************/

 Set Shutdowntime 60 ,               /* Reserve 60 seconds for CP      */
     Signal Shutdowntime 300      /* Default guest time is 300 seconds*/
```

> **Note:** **`Signal ShutdownTime 300`** permits any virtual machine that is sent a shutdown signal (sigkill) 300 seconds to complete the shutdown process before it is then forced off. Under most circumstances, this value is more than adequate. **`ShutdownTime 60`** permits any virtual machine that is sent a `FORCE` (forced log off) 60 seconds to quiesce before the forced log off happens.

f. Modify the comments for the lines that you changed, where suitable. You can use the text that is shown in Example 6-6 or enter comments of your own.

> **Important:** As in the C programming language, JavaScript, and Cascading Style Sheets (CSS), you must ensure that all comment strings are correctly enclosed between a pair of **`/*`** and **`*/`**, as shown in the following example:
>
> `/* DISABLE the following features */`

Example 6-6 Results of changes to the SYSTEM CONFIG file plus updated comments

```
/*********************************************************************/
/*                        Features Statement                         */
/*********************************************************************/
```

```
 Features ,
   Retrieve ,                      /* Retrieve options              */
     Default  20 ,                 /* Default.... default is 20     */
     Maximum  255 ,                /* Maximum.... default is 255    */
   MaxUsers noLimit ,              /* No limit on number of users   */
   Vdisk Userlim 144000 blocks,    /* Maximum vdisk allowed per user */
   Disconnect_Timeout off ,        /* Can be OFF, default is 15 min  */

   Enable ,                        /* Enable the following features  */
     New_Devices_Initialized_When_Added,  /* Make new devices online */
     Clear_TDisk   ,               /* Clear TDisks at IPL time */

   Disable ,                       /* Disable the following features */
     Set_Privclass ,               /* Disallow SET PRIVCLASS command */
     Auto_Warm_IPL ,               /* Prompt at IPL always          */
     Validate_Shutdown ,           /* Don't require system name     */
     STP_Timezone ,                /* STP feature is not used       */
     Paging_Alias ,           /* HyperPAV alias not used for paging  */
     Paging_HPF        /* High Performance FICON not used for paging */

/**********************************************************************/
/*                    Set Shutdown time periods                        */
/**********************************************************************/

 Set Shutdowntime 60 ,              /* Reserve 60 seconds for CP      */
     Signal Shutdowntime 300      /* Default guest time is 300
seconds*/
```

g. Save your changes by using the following commands and proceed:

```
====> file
====> cpsyntax system config f
```

Important: Always do CPSYNTAX after any modification on SYSTEM CONFIG.

The response is shown in the following example:

```
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

Tip: If your system is an SSI cluster, CPSYNTAX features a mandatory option that identifies on which LPAR you are doing the command. For example, our system identifier is:

```
System_Identifier LPAR LEPUS28 RDBKZVMG
```

Therefore, our **cpsyntax** command is:

```
cpsyntax system config f (lpar lepus28
```

Some of the **SYSTEM CONFIG** statements were changed from their default values. For more information, see *z/VM Version 7 Release 2: CP Planning and Administration*, SC24-6271.

6.5.2 Enabling and configuring virtual networking components

For each SSI member, set real device equivalency IDs (EQIDs) for the Open Systems Adapter (OSA) addresses to be used, and set the Media Access Control (MAC) address prefix. Real device mapping provides a means of identifying a device by EQID. This mapping ensures that virtual machines that are relocated by using the live guest relocation (LGR) feature continue to use the same or equivalent devices after a relocation is complete.

Your SYSTEM CONFIG file is still open in XEDIT from the tasks that were performed in 6.5.1, "Modifying features and optimizing parameter settings" on page 144. Complete the following steps:

1. Jump to the line that contains the string `STATUS OF DEVICES` and then, move up one line:

   ```
   ====> /status of devices
   ====> up 1
   ```

2. Use the XEDIT block copy function to copy the three lines that make up the heading of the `Status of Devices` stanza and paste them underneath as a new heading:

   a. In the prefix area of the current line, type `CC` over the numbers and press **Enter**. The `CC` turns red. Move down two lines and repeat this step.

   b. Move to the line after the last statement in the `Status of Devices` stanza (in our Example 6-7, the line containing the word `Sensed`), type `P` into the prefix area and press **Enter**.

   The entire heading is now duplicated. Example 6-7 shows the now duplicated heading.

   Example 6-7 Duplicated heading in SYSTEM CONFIG

   ```
   00273 /*****************************************************************/
   00274 /*                          Status of Devices                    */
   00275 /*****************************************************************/
   00276
   00277  Devices ,
   00278    Online_at_IPL   0000-FFFF,
   00279    Sensed          0000-FFFF
   00280
   00281 /*****************************************************************/
   00282 /*                          Status of Devices                    */
   00283 /*****************************************************************/
   ```

   c. Type over the string `Status of Devices` in the newly copied heading at the bottom with the string **`Virtual Network Configuration`**, as shown in Example 6-8.

   Example 6-8 Virtual network configuration heading

   ```
   /*****************************************************************/
   /*                          Status of Devices                    */
   /*****************************************************************/

    Devices ,
      Online_at_IPL   0000-FFFF,
      Sensed          0000-FFFF

   /*****************************************************************/
   /*                  Virtual Network Configuration                */
   /*****************************************************************/
   ```

3. Jump to the line that contains the string `Virtual Network Configuration` by entering the following command:

```
====> /virtual network configuration
```

To move down two lines and enable INPUT mode, run the following commands:

```
====> down 2
====> input
```

4. Enter the lines that are shown in Example 6-9. Press **Enter** after each line. Press **Enter** twice after the last line to return from INPUT mode.

Example 6-9 Virtual network configuration values for the first member of the SSI cluster

```
RDBKZVMG: BEGIN
          RDEV 2100-210F EQID OSA1SET1 TYPE OSA
          RDEV 2120-212F EQID OSA1SET1 TYPE OSA
          VMLAN MACPREFIX 02000A
          VMLAN LIMIT TRANSIENT 0
          DEFINE VSWITCH VSW1 RDEV 2100 2120 ETHERNET
          MODIFY VSWITCH VSW1 GRANT TCPIP
          DEFINE VSWITCH VSW2 ETHERNET
          MODIFY VSWITCH VSW2 GRANT TCPIP
RDBKZVMG: END
```

Example 6-10 on page 151 shows the entries for both LPARs together with key differences in bold.

Tip: If you are reading an electronic copy of this book, you can copy the entire block of text from Example 6-9 or Example 6-10 on page 151 and paste it directly into your 3270 XEDIT session while you are in INPUT mode. If you are manually entering the information, carefully enter the text from Example 6-9 and then, use the XEDIT block copy operation to duplicate the entire block of text for each extra LPAR in your cluster.

Consider the following points:

– After you enter the six lines of text manually the first time, you can use the XEDIT block copy function to duplicate the block and then update the block for each member of your cluster.

– The **VMLAN MACPREFIX** statement sets the first three bytes of the Media Access Control (MAC) address that was created for each virtual network interface card (NIC). Obtain these values from the planning worksheet. In this example, ***02000A*** and ***02000B*** are used.

Note: Ensure that you double-check your work to avoid creating identical MAC addresses.

– The **VMLAN LIMIT TRANSIENT 0** statement prevents dynamic definition of Guest LANs by class G users that interfere with the ability to relocate.

– The **DEFINE VSWITCH** statements define a pair of MAC-based Ethernet virtual switches (VSWITCHES). MAC Ethernet VSWITCHES are sometimes referred to as *LAYER 2 VSWITCHES*. Modify the two starting addresses of the OSA triplets to those addresses that you specified in your planning worksheet.

> **Important:** For setting the `VMLAN MACPREFIX` value, *IBM z/VM CP Planning and Administration*, SC24-6271, states the following information:
>
> > "In an SSI cluster, system-defined locally administered MAC addresses are created by using the prefix value that is specified on the `MACPREFIX` operand. The `MACPREFIX` value must be different for each member of the cluster. The default value is 02*xxxx*, where *xxxx* is the member's slot number on the SSI statement. If the `MACPREFIX` value is explicitly defined, the `VMLAN` statement must be qualified for the member to which it applies. Therefore, if a `VMLAN` statement with the `MACPREFIX` operand is retained from the non-SSI system or created in this step, it must be qualified for member `VMSYS01`."

Example 6-10 Virtual network configuration additions to the SYSTEM CONFIG file

```
/*********************************************************************/
/*                    Virtual Network Configuration                 */
/*********************************************************************/
RDBKZVMG: BEGIN
          RDEV 2100-210F EQID OSA1SET1 TYPE OSA
          RDEV 2120-212F EQID OSA1SET1 TYPE OSA
          VMLAN MACPREFIX 02000A
          VMLAN LIMIT TRANSIENT 0
          DEFINE VSWITCH VSW1 RDEV 2100 2120 CONTROLLER * ETHERNET
          MODIFY VSWITCH VSW1 GRANT TCPIP
          DEFINE VSWITCH VSW2 CONTROLLER * ETHERNET
          MODIFY VSWITCH VSW2 GRANT TCPIP
RDBKZVMG: END
RDBKZVMH: BEGIN
          RDEV 2100-210F EQID OSA1SET1 TYPE OSA
          RDEV 2120-212F EQID OSA1SET1 TYPE OSA
          VMLAN MACPREFIX 02000B
          VMLAN LIMIT TRANSIENT 0
          DEFINE VSWITCH VSW1 RDEV 2100 2120 CONTROLLER * ETHERNET
          MODIFY VSWITCH VSW1 GRANT TCPIP
          DEFINE VSWITCH VSW2 CONTROLLER * ETHERNET
          MODIFY VSWITCH VSW2 GRANT TCPIP
RDBKZVMH: END
```

5. Save your changes and quit XEDIT with the **`FILE`** subcommand:

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

6.5.3 Using CPSYNTAX to validate the modified system configuration file

The `CPSYNTAX` utility attempts to catch incorrect and unrecognized statements in the `SYSTEM CONFIG` file. It does not attempt to identify problems between statements or valid but duplicate operands on a single statement. CPSYNTAX does not ensure that it finds all configuration file changes that lead to a problem during system IPL; you still must approach edits of `SYSTEM CONFIG` carefully to ensure that you do not make mistakes.

During system IPL, configuration file post-processing routines perform more checking of the data that is specified in the configuration file. The IPL of a second-level system to check the configuration file is recommended for a more thorough test and you might find problems that CPSYNTAX did not.

However, a second-level IPL does not eliminate environmental factors of the target first-level system that the file is intended for, and still might not find all problems that relate to the system configuration file.

Complete the following steps:

1. Test the changes that are made to `SYSTEM CONFIG` with the **CPSYNTAX** command, which is on the `MAINT 193` disk. The **CPSYNTAX** command must be run one time for each member of the SSI cluster by using the **LPAR** option to the command:

```
===> access 193 G
===> cpsyntax system config F (LPAR A09
   CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
===> cpsyntax system config F (LPAR A0A
   CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

   Pay attention to the output. If you get any syntax errors, fix them before you proceed.

2. Release, *but do not detach,* the `MAINT 193` disk:

```
===> release G
```

3. Release and detach the `PMAINT CF0` disk:

```
===> release F (detach
DASD 0120 DETACHED
```

The `SYSTEM CONFIG` file is now ready for use. Log off from `MAINT720`.

6.5.4 Initializing the allocated DASD for z/VM Service data

Perform these steps to initialize the DASD that was selected to serve as the repository disk for the z/VM Service data. This data will be used by VMSES/E, which is the component of z/VM that applies Service to your system.

1. From the HMC Integrated 3270 Console, log on as **MAINT720**. The default password for you is provided in the information that you obtained from IBM with your z/VM order.

   You see output similar to the following output:

```
LOGON MAINT720
z/VM Version 7 Release 2.0, Service Level 2001 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0004 RDR,   NO PRT,   NO PUN
LOGON AT 18:36:23 EDT TUESDAY 09/29/20
GRAF L0005 LOGON  AS  MAINT    USERS = 16    FROM 9.60.70.32
z/VM V7.2.0    2020-06-26 09:03

DMSACP723I B (5E5) R/O
DMSACP723I D (51D) R/O
DMSACP723I E (551) R/O
```

2. If you do not see the Conversational Monitor System (CMS) `Ready;` prompt, press Enter and it will appear.

3. By using the value from the planning worksheet, attach the DASD to MAINT720 by using its real device address. In the example environment that was used to author this book, that device address is 1564.

```
===> attach 1564 to *
===> cpfmtxa 1564
ENTER FORMAT, ALLOCATE, LABEL, OWNER OR QUIT:
```

```
format
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 1564 OR QUIT:
0-END
ENTER THE VOLUME LABEL FOR DISK 1564:
VV1564
CPFMTXA:
FORMAT WILL ERASE CYLINDERS 000000000-000003338 ON DISK 1564
DO YOU WANT TO CONTINUE? (YES | NO)
yes
HCPCCF6209I INVOKING ICKDSF.
   ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
   CONSOLE
   ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
   CONSOLE
   ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0 ...

   ENTER INPUT COMMAND:
     CPVOL FMT MODE(ESA) UNIT(1564) VOLID(VV1564) NOVFY NFILL -
   ENTER INPUT COMMAND:
    RANGE(0,3338)
   ICK00700I DEVICE INFORMATION FOR 1564 IS CURRENTLY AS FOLLOWS:
             PHYSICAL DEVICE = 3390
             STORAGE CONTROLLER = 3990
             STORAGE CONTROL DESCRIPTOR = E9
             DEVICE DESCRIPTOR = 0A
             ADDITIONAL DEVICE INFORMATION = 4A001F3C
             TRKS/CYL = 15, # PRIMARY CYLS = 3339
   ICK04000I DEVICE IS IN SIMPLEX STATE
   ICK00091I 1564 NED=002107.900.IBM.75.0000000AKAZ1
   ICK091I   1564 NED=002107.900.IBM.75.0000000AKAZ1
   ICK03020I CPVOL WILL PROCESS 1564 FOR VM/ESA MODE
   ICK03090I VOLUME SERIAL = VV1564
   ICK03022I FORMATTING THE DEVICE WITHOUT FILLER RECORDS
   ICK03011I CYLINDER RANGE TO BE FORMATTED IS 0 - 3338
   ICK003D REPLY U TO ALTER VOLUME 1564 CONTENTS, ELSE T
   U
   ICK03000I CPVOL REPORT FOR 1564 FOLLOWS:

   FORMATTING OF CYLINDER 0 STARTED AT: 19:37:30
            FORMATTING OF CYLINDER 100 ENDED AT: 19:37:30
   ...

             VOLUME SERIAL NUMBER IS NOW = VV1564

          CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
          TYPE     START      END        TOTAL
          ----     -----      ---        -----
PERM     0          3338       3339

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

ENTER INPUT COMMAND:
 END

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

```
ENTER ALLOCATION DATA
TYPE CYLINDERS
.................
PERM 0 END
END
```

4. Remain logged on as MAINT720 and proceed.

6.5.5 Service-level validation and subscribing to service notifications

Perform these steps to ensure that the initial recommended service upgrade (RSU) was installed, and to be automatically notified of high priority service releases from IBM:

1. You are still logged on as MAINT720 through the HMC Integrated 3270 Console from the previous step.
2. Issue the **QUERY CPLEVEL** command to see the RSU level. In this example, it is **1401**:

   ```
   ===> query cplevel
   z/VM Version 7 Release 2.0, service level 2001 (64-bit)
   Generated at 07/29/20 16:50:40 EDT
   IPL at 09/26/20 20:13:22 EDT
   ```

3. See the the RSU page for z/VM to determine the latest available RSU:
   a. Under the column that is labeled RSU Content, click **ZVM720** to see the details of the latest RSU.
   b. At the top of the page, look for the following text:

      ```
      This file contains APAR/PTFs included on the z/VM Version 6, Release 3
      Modification 0, 2001RSU tape.
      ```

      In this example, **2001RSU** is the newest RSU that is available for z/VM 6.3.0.
4. If they do not match, download and apply the latest RSU now before you proceed further. Instructions to download the latest RSU are in 8.3, “Applying a recommended service upgrade” on page 259.
5. Subscribe to the Service News and Red Alerts pages:
   a. Navigate to the Service News page.
   b. Click **notify** in the left navigation menu for the site and complete the subscription form to enable automatic email notification for service news, such as the availability date of a new RSU:

      Action: **Enroll**

      ```
      Your e-mail address: ...
      File: /service/news
      ```

   c. Navigate to the Red Alerts page.
   d. Click **notify** in the left navigation menu and complete the form to enable automatic email notification when a critical fix is available:

      Action: **Enroll**

      ```
      Your e-mail address: ...
      File: /service/redalert
      ```

Note: For more information about z/VM Service, see Chapter 8, “Servicing z/VM” on page 255

A z/VM 7.2 SSI cluster is now available. Proceed to begin the configuration of your new SSI cluster.

6.6 Enabling and configuring DirMaint

DirMaint helps you to manage the USER DIRECT file, which holds all information about the z/VM user ID and makes user ID and resource administration easier.

It is recommended that you enable DirMaint and configure it before enabling and configuring RACF.

> **Note:** For more information about DirMaint and its configuration, see Chapter 10, “DirMaint, RACF-connector, and SMAPI” on page 305.

6.7 Enabling and configuring RACF

This section assumes that a new RACF database is being created. For migrating a RACF database, see the *Program Directory for RACF Security Server for z/VM*.

> **Note:** This section also assumes that DirMaint, RACF-connector, and SMAPI were configured according to the information in Chapter 10, “DirMaint, RACF-connector, and SMAPI” on page 305.

To configure RACF on a new z/VM 7.2 system, complete the following steps. The first five steps are performed before RACF is started. Steps 6 and 7 put RACF into production:

1. Creating the RACF RPIDIRCT command file.
2. Customizing SMF.
3. Copying the RACF databases.
4. Setting up the AUTOLOG1 and AUTOLOG2 virtual machines.
5. Enabling RACF.
6. Putting RACF into production on all members.
7. (The step is performed after RACF is in production:) Configuring SMAPI to work with RACF, as described in Chapter 10, “DirMaint, RACF-connector, and SMAPI” on page 305.

Resource Access Control Facility (RACF) is a security server that is available for z/OS and z/VM. The command interface is similar on both systems; however, the functionality on z/VM is limited to the resources that are available to z/VM.

You must activate the management of resources before RACF takes over. For resources, such as VSWITCHES, the access control is taken over by RACF. Running commands, such as **`set vswitch vsw1 grant`** <<*userid*>>, does not set anything.

After you activate RACF, if you encounter access problems, it is a preferred practice to review the operator console. Often, the access issue is reported on the operator console.

Important: If you plan to enable RACF, consider the following points:

- You must decide on the set of activities that you want to audit, and whether audit is always on for those activities or only on demand. It will be necessary to **LINK** and **ACCESS** the active System Management Facilities (SMF) disk to see how fast it is filling. In a Linux farm, most of the activity will be the system programmers' and system administrators' activities.
- If both the primary and secondary SMF minidisks unexpectedly become full, no more audit records can be recorded, even though security-relevant events can continue to occur. Naturally, any such loss of audit records is unacceptable in a secure system. The `SEVER YES` setting in the `SMF CONTROL` file instructs RACF to *sever* when this situation happens. This setting ensures "If it didn't get written down, it didn't happen," which is an excellent policy if you are being cross-examined on the witness stand (possibly as the accused) in a data theft case.
- The SMF log disks need to be sized to hold an audit log that has all of the data for a single archive interval. That is, if `RACFSMF` is logged on once a day, the SMF disks need to be large enough to hold one day's worth of data. (Because two disks are available, it can hold double that amount per day.)
- The `RACFSMF 192` archive disk needs to be large enough to hold '*n*' archives, where '*n*' is your defined value, as a safety mechanism. The oldest files need to be erased as required to make room for the latest archive. Warning: As shipped, RACFSMF does not wrap; it simply sends a message to `OPERATOR` when the disk is 80% full.
- You must modify RACFSMF to send the newly archived file to a more permanent location. It can use File Transfer Protocol (FTP) to send it, put it in SFS, SENDFILE to IBM MVS, dump to tape, or IBM FlashCopy® the 192 to the next in a series of disks. It is useful to have several pre-packaged skeleton activities in SMFPROF.

6.7.1 Creating the RACF RPIDIRCT command file

To set up the initial RACF database, a set of RACF commands is constructed from the user directory source file, then modified later. The **RPIDIRCT EXEC** helps you to migrate the user directory data to a RACF database. It translates directory statements into RACF commands and puts them in an output file named `RPIDIRCT SYSUT1`.

To create `RPIDIRCT SYSUT1` for later use with **RPIDIRCT**, complete the following steps:

1. Log on to **MAINT** on the first SSI member.
2. Link the `7VMRAC20 191` disk read/write and access it as file mode `F`:

   ```
   ===> link 7VMRAC20 191 1191 mr
   ===> acc 1191 f
   ```

3. Link the `7VMRAC20 505` disk read/write and access it as file mode `G`:

   ```
   ===> link 7VMRAC20 505 1505 mr
   ===> acc 1505 g
   ```

4. If you are using DirMaint, get the current user directory with passwords with the **DIRMAINT USER WITHPASS** command:

   ```
   ===> dirm user withpass
   DVHXMT1191I Your USER request has been sent for processing to DIRMAINT
   DVHXMT1191I at POKDEV62.
    DVHREQ2288I Your USER request for MAINT at * has been accepted.
   ```

```
RDR FILE 0004 SENT FROM DIRMAINT PUN WAS 0005 RECS 4539 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your USER request for MAINT at * has completed; with RC = 0.
```

Receive the file onto the 7VMRAC20 191 disk (F). In this example, the reader file was the number 4 that was noted from the previous command output:

```
===> receive 4 = = f
File USER WITHPASS F0 created from USER WITHPASS A0 received from DIRMAINT at
POKDEV62
```

5. If you are *not* using DirMaint, manually copy USER DIRECT:

```
===> copy USER DIRECT C = = F
```

6. Create the RPIDIRCT SYSUT1 file from the user directory with the **RPIDIRCT** command. Enter n to the question of changing the default group ID. This response allows RACF to give all of the existing virtual machines access to the resources that they currently have.

   You might want to issue a **#CP TERM MORE 0 0** because many panels of output will scroll by:

   If you used DirMaint to get the user directory, use this command:

```
===> rpidirct user withpass f
```

   If you manually edited a copy of the USER DIRECT file, run the following command:

```
===> rpidirct user direct f
Both commands will show the following
Output defaulted to "A" disk.
  Default group ID = SYS1.
  Would you like to change this default?
  Enter Y/N
n
  Default group ID = SYS1.

PROFILE IBMDFLT
...
PROFILE TCPCMSU
...
```

   After several messages, you will see:

```
**********  3936 Directory records processed  *********
***************  RPIDIRCT SYSUT1 CREATED  **************
```

7. Copy the newly created RPIDIRCT SYSUT1 file so that you have a reference:

```
===> copy rpidirct sysut1 a = sysuorig =
```

8. In the new RPIDIRCT SYSUT1 file, remove all of the lines with the text VMBATCH. A generic VMBATCH profile will be created shortly. All lines can be deleted with the **ALL** subcommand and the prefix command **d*** (hidden lines will not be deleted):

```
===> x rpidirct sysut1
====> all /VMBATCH/
====> top
d*=== * * * Top of File * * *
===== --------------------  22  line(s) not displayed --------------------
===== RDEFINE VMBATCH $ALLOC$ OWNER($ALLOC$) UACC(NONE)
...
...
====> all
```

   All lines with VMBATCH are now deleted.

9. Add the following lines to the bottom of the RPIDIRCT SYSUT1 file:

```
====> bot
====> a 4
setropts generic(vmbatch) gencmd(vmbatch)
rdefine vmbatch ** uacc(none)
permit ** class(vmbatch) id(ftpserve vmnfs dirmsat dirmsat2) acc(control)
setropts classact(vmbatch vmmdisk vmcmd vmlan surrogat)
====> file
```

> **Notes:**
>
> - The first two lines make VMBATCH a generic class.
> - The third line permits the FTP, Network File System (NFS), and DirMaint satellite servers to the VMBATCH class. The number of DIRMSAT* entries needs to correspond to the number of members in the SSI (for example, if you use a four-member SSI, add DIRMSAT3 and DIRMSAT4). Permitting the servers to the VMBATCH class will allow them to use the alternate user ID function.
> - For more information about protecting this function, see the “Protecting Alternate User IDs” section of the *z/VM RACF Security Server Auditor’s Guide,* SC24-6212.
> - The fourth line activates the classes VMBATCH, VMMDISK, VMCMD, VMLAN, and SURROGAT.

10. Move the file to the 7VMRAC20 191 disk (F) by using the following commands:

```
===> copy rpidirct sysut1 a = = f
===> erase rpidirct sysut1 a
```

The modified RPIDIRCT SYSUT1 file is now on the 7VMRAC20 191 disk.

6.7.2 Customizing SMF

One of the reasons that you run RACF on your z/VM system is to be able to audit who is doing what on the system. The audit records must be managed through the RACFSMF virtual machine.

To create a **PROFILE EXEC** for the RACFSMF virtual machine, complete the following steps:

1. Link the RACFSMF 191 disk read/write and access it as file mode H:

```
===> link racfsmf 191 2191 mr
===> acc 2191 h
```

2. Copy the sample profile SMFPROF EXEC to the RACFSMF 191 disk (H) as the file PROFILE EXEC:

```
===> copy smfprof exec g profile = h
```

3. Edit the **PROFILE EXEC** and change the value of Smffreq to **AUTO** and Smfswtch to **NO**:

```
===> x profile exec h
====> /Smfdisk
====> =
...
Smfdisk  = 192
Smfpct   = 80
Smfinfo  = 'OPERATOR'     /* Default message receiver        @VA45455*/
Smffreq  = 'AUTO'         /* Valid values: DAILY, WEEKLY, MONTHLY,    */
                          /*               AUTO              @VA45455*/
Smfday   = 'MONDAY'       /* Valid values: SATURDAY - FRIDAY  @VA45455*/
Smfswtch = 'NO'           /* Valid values: YES NO            @VA45455*/
```

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

Note: These changes to the `RACFSMF PROFILE EXEC` will archive SMF data only when the SMF disk is full. If your site requires archiving regularly, you can use this EXEC and xautolog the user at each interval.

For more information, see the chapter, "Processing Audit Records on z/VM", in *z/VM RACF Security Server Auditor's Guide,* SC24-6305.

The **PROFILE EXEC** is now configured for the `RACFSMF` virtual machine.

Modifying the SMF CONTROL file

To set `SEVER YES` in the `SMF CONTROL` file on the `RACFVM 191` disk, complete the following steps:

1. Link to the `RACFVM 191` disk read/write and access it as file mode `I`:

   ```
   ===> link racfvm 191 3191 mr
   ===> acc 3191 t
   ```

2. Edit the `SMF CONTROL` file and change `SEVER NO` to **SEVER YES**:

   ```
   ===> x smf control t
   ====> prefix off
   * * * Top of File * * *
   CURRENT 301 K PRIMARY 301 K SECONDARY 302 K 10000 VMSP CLOSE 001 SEVER YES 0
   RAC
   ====> file
   Setting this value to YES will cause RACF to disconnect from the control
   program (CP) if the SMF disks are full.
   ```

   Note: When RACF is disconnected from CP, users will be unable to log on. To fix the full SMF disk, you will need to log on through OPERATOR by using its CP password and IPL CMS. You can copy the SMF records and then clear out the SMF records. Then, restart RACFVM.

3. Copy the modified SMF CONTROL file to the RACFSMF 191 (H) disk:

   ```
   ===> copy smf control t = = h (rep
   ```

4. Link the `RACMAINT 191` disk read/write and access it as file mode `J`:

   ```
   ===> link racmaint 191 4191 mr
   ===> acc 4191 j
   ```

5. Copy the modified `SMF CONTROL` file to the `RACMAINT 191` disk (J) with the **REPLACE** option:

   ```
   ===> copy smf control t = = j (rep
   ```

6. Log off from `MAINT`.

The SMF configuration of RACF is now complete.

6.7.3 Copying the RACF databases

In an SSI cluster, the RACF database must be shared among all members. If you are installing RACF in a single z/VM logical partition (LPAR) only, you can skip this section, which consists of the following subsections:

- Copying the RACFVM 200 and 300 minidisks.
- Changing RACFVM to shared disks.
- Modifying the RACMAINT identity.
- Defining the shared disks in the SYSTEM CONFIG file.

> **Note:** When RACF is installed in a single-member or multiple-member z/VM single system image (SSI) environment, the RACF database must be configured as being shared.

Copying the RACFVM 200 and 300 minidisks

To copy the `RACFVM 200` and `300` minidisks to the volumes that will be shared, complete the following steps:

1. Log on to the first SSI member as `MAINT`:

   ===> logon maint by ibmvm1

   > **Important:** If your SSI is on LPARs at the first level, you must use real volumes for the `200` and `300` RACF database, they cannot be minidisks. Use the smallest volumes that you can get because the RACF database does not need many cylinders, even mod-3 is more than enough in most cases. It is not recommended to use volumes with more than 32,760 cylinders.

2. Attach the DASD volumes that will be shared:

```
===> q 103B 113B
DASD 103B NW103B  , DASD 113B NW113B
===> att 103B 113B *
0200 0300 ATTACHED TO MAINT
```

3. Change the label with the **`CPFMTXA`** command so that the second character is "`R`" to signify RACF. The second character must not be "`M`" for minidisk or it will be attached to `SYSTEM` at z/VM IPL time:

```
===> cpfmtxa 103b jr103b label
...
VOLUME SERIAL NUMBER IS NOW = JR103B

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...
===> cpfmtxa 113b jr113b label
...
VOLUME SERIAL NUMBER IS NOW = JR113B

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...
```

4. Link to the `RACFVM 200` and `RACFVM 300` disks read-only with the **`VMLINK`** command:

```
===> vmlink racfvm 200
DMSVML2060I RACFVM 200 linked as 0120 file mode Z
===> vmlink racfvm 300
DMSVML2060I RACFVM 300 linked as 0121 file mode X
```

The virtual device addresses of the linked disks are 120 (for RACFVM 200) and 121 (for RACFVM 300).

5. Copy the RACFVM 200 disk (120) to the 103B volume with the **DDR** command and the following subcommands:

```
===> ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
====> sysprint cons
ENTER:
====> in 120 3390
ENTER:
====> out 103b 3390
ENTER:
copy 0 to 16
HCPDDR711D VOLID READ IS RACF
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
ENTER NEXT EXTENT OR NULL LINE
ENTER:

HCPDDR711D VOLID READ IS JR103B
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
COPYING   RACF
COPYING DATA  06/10/13 AT 18.49.57  GMT FROM RACF   TO JR103B
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP         START        STOP
          0         16             0          16
END OF COPY Enter
END OF JOB
```

6. Copy the RACFVM 300 disk (121) to the 113B volume with the **DDR** command and the following subcommands:

```
===> ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
====> sysprint cons
ENTER:
====> in 121 3390
ENTER:
====> out 113B 3390
ENTER:
====> copy 0 to 16
HCPDDR711D VOLID READ IS RACFBK
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
ENTER NEXT EXTENT OR NULL LINE
ENTER:

HCPDDR711D VOLID READ IS JR113B
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
COPYING   RACFBK
COPYING DATA  06/10/13 AT 18.53.36  GMT FROM RACFBK TO JR113B
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
```

```
      START      STOP        START       STOP
          0        16            0         16
END OF COPY
ENTER:
Enter
END OF JOB
```

The contents of the RACF data sets on the RACFVM 200 and 300 minidisks were copied to the real devices (at addresses 103B and 113B in this example).

Changing RACFVM to shared disks

Note: Non-SSI z/VM systems can remain by using minidisks and shared disks.

Now that the 200 and 300 minidisks from one of the SUBCONFIGs of RACFVM were copied to the DASD volumes that will be shared, these new disks can replace the individual minidisks.

Complete the following steps:

1. Get the user directory entry of the RACFVM-1 SUBCONFIG:

   ```
   ===> dirm for racfvm-1 get
   ...
   ```

2. Receive the file from the reader.
3. Comment out the 200 and 300 disks:

   ```
   ===> x racfvm-1 direct
   SUBCONFIG RACFVM-1
    LINK MAINT 0190 0190 RR * CMS system disk
    LINK MAINT 019D 019D RR * help disk
    LINK MAINT 019E 019E RR * Product code disk
    MDISK 191 3390 1568 009 JV1033 MR READ     WRITE    MULTIPLE
   * MDISK 200 3390 1551 017 JV1033 MW READ     WRITE    MULTIPLE
    MDISK 490 3390 1577 070 JV1033 MR READ     WRITE    MULTIPLE
    MDISK 305 3390 1647 136 JV1033 MR READ     WRITE    MULTIPLE
   * MDISK 300 3390 1783 017 JV1033 MW READ     WRITE    MULTIPLE
    MDISK 301 3390 1800 007 JV1033 MR READ     WRITE    MULTIPLE
    MDISK 302 3390 1807 007 JV1033 MR READ     WRITE    MULTIPLE
   ===> file
   ```

4. Replace the RACFVM-1 SUBCONFIG definition:

   ```
   ===> dirm for racfvm-1 rep
   ...
   ```

5. Repeat the previous steps for all other members in the SSI cluster. In this example, only the RACFVM-2 SUBCONFIG also must be modified.
6. Get the user directory entry of the IDENTITY RACFVM:

   ```
   ===> dirm for racfvm get
   ...
   ```

7. Receive the file from the reader.
8. Add the following MDISK entries for 200 and 300:

   ```
   ===> x racfvm direct
   IDENTITY RACFVM   RACFVM     20M   20M ABCDEGH
    BUILD ON LEFT630 USING SUBCONFIG RACFVM-1
   ```

```
 BUILD ON RIGHT630 USING SUBCONFIG RACFVM-2
* BUILD ON @@member3name USING SUBCONFIG RACFVM-3
* BUILD ON @@member4name USING SUBCONFIG RACFVM-4
 IUCV *RPI PRIORITY MSGLIMIT 100
 IUCV ANY PRIORITY MSGLIMIT 50
 IUCV ALLOW MSGLIMIT 255
 ACCOUNT SYSTEMS
 MACH XA
 IPL 490 PARM AUTOCR
 OPTION QUICKDSP MAXCONN 300
 CONSOLE 009 3215 T OPERATOR
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
* Add minidisks 200 and 300 for a shared RACF database
 MDISK 200 3390 DEVNO 103B      MWV READ     WRITE    MULTIPLE
 MDISK 300 3390 DEVNO 113B      MWV READ     WRITE    MULTIPLE
...
```

The DEVNO operand on the MDISK statement specifies a full-pack minidisk. It allows CP to not depend on the volume labels of the disks.

9. Replace the RACFVM SUBCONFIG definition:

```
===> dirm for racfvm rep
...
 DVHREQ2289I Your REPLACE request for RACFVM at * has completed; with
 DVHREQ2289I RC = 0.
```

Watch for a return code of 0.

The RACFVM virtual machine now references the two shared DASD volumes.

Note for users who wants to migrate RACF databases from Mod3 to Mod9: You can use the procedure that is described in this section by copying the disks with DDR or FlashCopy and updating the new addresses on the RACFVM directory to point to the new DASD.

Modifying the RACMAINT identity

The IDENTITY RACMAINT has link modes to the RACFVM 200 and 300 minidisks of MR. They must be changed to **MW** to share the RACF database. Complete the following steps:

1. Get the user directory entry of the RACMNT-1 SUBCONFIG:

```
===> dirm for racmnt-1 get
...
```

2. Receive the file from the reader.
3. For the RACMAINT SUBCONFIGs, change the link modes to the RACFVM 200 and 300 disks from MR to **MW**. First, the RACMNT-1 SUBCONFIG is changed:

```
===> x racmnt-1 direct
 SUBCONFIG RACMNT-1
  LINK MAINT 0190 0190 RR * CMS system disk
  LINK MAINT 019D 019D RR * help disk
  LINK MAINT 019E 019E RR * Product code disk
  LINK 7VMRAC20 590 490 MR
  LINK 7VMRAC20 505 305 MR
  LINK 7VMRAC20 29E 29E RR
```

```
  LINK 7VMRAC20 191 192 RR
  LINK RACFVM 200 200 MW
  LINK RACFVM 300 300 MW
  LINK RACFVM 301 301 MR
  LINK RACFVM 302 302 MR
===> file
```

4. Replace the user directory entry:

```
===> dirm for racmnt-1 rep
...
```

5. Repeat the previous steps for all other members in the SSI cluster. In this example, two-member SSI cluster, only the RACMNT-2 SUBCONFIG needed to be modified.

The RACF database can now be shared on the volumes at real device addresses 103B and 113B.

Defining the shared disks in the SYSTEM CONFIG file

To define the RACF database DASD to CP as devices that can be shared concurrently between real systems, you must add the RDEVICE statements to the SYSTEM CONFIG file.

Complete the following steps:

1. Verify that you are logged on as MAINT.
2. Access the PMAINT CF0 disk read/write. Use the **LINK** command with the multi-read (**MR**) parameter:

```
===> link pmaint cf0 cf0 mr
```

3. Use the **ACCESS** command to access it as F:

```
===> acc cf0 f
```

4. Make a copy of the working SYSTEM CONFIG file:

```
===> copy system config f = confwrks = (rep
```

5. Edit the original file:

```
===> x system config f
```

6. Add two lines at the bottom that specify that the primary and backup RACF database disks are shared:

```
====> bot
====> a 3
...
/* Define RACF primary and backup databases as shared */
rdevice 103B type dasd shared yes /* RACF primary database */
rdevice 113B type dasd shared yes /* RACF backup database */
```

7. Verify the syntax of the file with your LPAR names as the parameter:

```
===> acc 193 g
===> cpsyntax system config f (lpar a02
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
===> cpsyntax system config f (lpar a2e
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

8. Release and detach the PMAINT CF0 (F) disk:

```
===> rel f (det
DASD 0CF0 DETACHED
```

It is also a requirement that CP does not cache data on the RACF database disks in the minidisk cache. Minidisk cache (MDC) is turned off as a result of specifying the DASD as shared in the system configuration file.

The RACF database and backup database now are shared in the SSI cluster.

6.7.4 Setting up the AUTOLOG1 and AUTOLOG2 virtual machines

At z/VM IPL time, the AUTOLOG1 virtual machine normally starts all necessary systems and virtual machines in its **PROFILE EXEC**. When RACF is running, the RACFVM virtual machine must be started first, or other virtual machines will not be able to log in. After the RACF environment is initialized, RACFVM starts the AUTOLOG2 virtual machine, which then starts the remaining servers for the system as AUTOLOG1 normally does.

Therefore, the **PROFILE EXEC** needs to be copied from AUTOLOG1 to AUTOLOG2, then modified to start RACFVM.

Complete the following steps:

1. Verify that you are logged on as MAINT on the first member.
2. Link the AUTOLOG1 and AUTOLOG2 191 disks read/write:

   ```
   ===> link autolog1 191 1191 mr
   ===> link autolog2 191 2191 mr
   ```

3. Access the two disks as file modes F and G:

   ```
   ===> acc 1191 f
   ===> acc 2191 g
   ```

4. Copy the PROFILE EXEC from AUTOLOG1 to AUTOLOG2:

   ```
   ===> copy profile exec f = = g
   ```

5. Edit the PROFILE EXEC on the AUTOLOG1 191 disk and replace the entire contents with the following contents to start RACFVM first:

   ```
   ===> x profile exec f
   ===> del *
   ===> top
   ===> a 6
   /*******************************************************************/
   /*  AUTOLOG1 PROFILE EXEC                                          */
   /*******************************************************************/
   Address Command
   "CP XAUTOLOG RACFVM"
   "CP LOGOFF"
   ====> file
   ```

6. Repeat these steps on all other SSI members in the cluster.

The AUTOLOG1 virtual machine is now configured. Start RACF (the RACFVM virtual machine). RACF will then start AUTOLOG2 to complete the bootstrapping of the z/VM system.

6.7.5 Enabling RACF

To enable RACF, complete the following steps:

> **Note:** If you are running in a non-SSI z/VM system, go directly to step 1b without issuing a shutdown.

1. Shut down all other members, except the first SSI node. In this example, SSI member 2 was shut down:

   ```
   ===> shutdown
   ...
   ```

   a. Log on as `MAINT720` on the first SSI member.

   b. Issue the following **SERVICE** command to enable RACF. This step must be performed on only one member. Several panels scroll by:

   ```
   ===> service racf enable
   several msgs after, you will see:
   ...
   VMFSRV1233I The following products have been serviced.
   VMFSRV1233I CPSFS RACFSFS
   VMFSRV2264I Restoring prior system environment using saved access/minidisk
               information
   VMFSET2760I VMFSETUP processing started for ENVRESTORE
               SERVICEEXEC20201001194312
   VMFSET2204I Linking MAINT720 0490 as 0490 with link mode RR
   VMFSET2204I Linking MAINT720 0493 as 0493 with link mode RR
   VMFSET2204I Linking PMAINT 0551 as 0551 with link mode RR
   VMFSET2204I Linking RACFVM 0300 as 0121 with link mode RR
   VMFSET2204I Linking RACFVM 0200 as 0120 with link mode RR
   VMFSET2760I VMFSETUP processing completed successfully (RC=0)
   VMFSRV2760I SERVICE processing completed successfully (RC=0)
   ```

   RACF is now enabled on the `CF2` disk. This disk is now on the release 1 volume in z/VM 7.2.

2. Shut down the first SSI member:

   ```
   ===> shutdown
   ...
   ```

RACF is now be enabled. Shut down all members and the SSI.

> **Note:** If you are on a non-SSI z/VM system, you shutdown the z/VM system.

6.7.6 Putting RACF into production on all members

> **Attention:** The **PUT2PROD** command must be run on each member of the SSI. Start with the first member. Perform all five of the following subsections on the first member. If you are in an SSI, you perform later only the *first and last* subsections on the other members:
>
> 1. IPLing the member and start RACMAINT.
> 2. Configuring the initial RACF database.
> 3. Enabling RACF-connector on DirMaint on the first member.
> 4. Setting the DirMaint use of the reader with RACF on the first member.
> 5. Putting RACF into production.

IPLing the member and start RACMAINT

You must IPL each member of the SSI and start RACMAINT. Complete the following steps:

1. Start an Integrated 3270 Console for the member.
2. IPL the member from the Hardware Management Console (HMC) from the real device address “Res volume”.

   The STAND ALONE PROGRAM LOADER (SAPL) window as shown in Figure 6-2 on page 167 opens on the Integrated 3270 Console.
3. Change the *Device Number* to the device number of **Release Volume 1** (volume label VMFRL1 in this case), not the “Res volume” (volume label VMFRES) that normally IPLs. In our example that is shown in Figure 6-2 on page 167, we show real device address ***1136***. Press **F10** to IPL, which loads the **CPLOAD MODULE** from the CF2 disk, where RACF is enabled.

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 2.0

DEVICE NUMBER:   1136     MINIDISK OFFSET:   39        EXTENT:  1

MODULE NAME:     CPLOAD     LOAD ORIGIN:       1000

 --------------------------------IPLPARAMETERS------------------------------
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=1036



----------------------------------COMMENTS---------------------------------




---------------------------------------------------------------------------



 9= FILELIST  10= LOAD  11= TOGGLE EXTENT/OFFSET
```

Figure 6-2 STAND ALONE PROGRAM LOADER window

4. Supply the NOAUTOLOG parameter so that the PROFILE EXEC on AUTOLOG1 is not run and RACFVM is not started:

```
16:30:25 Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable)  (NODIRect)
16:30:25       (NOAUTOlog)) or (SHUTDOWN)
noautolog
...
```

5. Continue to IPL the member. When the IPL process completes, you are logged on as OPERATOR. Start the virtual machine RACMAINT. You see messages that indicate that the 200 and 300 disks are read/write. If you see errors about them, fix the problem:

```
===> xautolog racmaint
...
```

RACF is now running on the SSI member with a skeleton database.

> **Note:** If you completed the next three sections on the first SSI member, proceed to "Putting RACF into production" on page 172.

Configuring the initial RACF database

The following steps must be performed only once to populate and customize the RACF database:

1. On the first SSI member, disconnect from OPERATOR:

```
===> disc
```

2. Log on to IBMUSER with a password of **SYS1**, which is a default virtual machine that is created for RACF configuration.
3. You see a message that the password expired. Reset the password by entering the new password twice. Separate the passwords with a forward slash (/). You see resource errors, which are expected:

```
LOGON IBMUSER
RPIMGR042I PASSWORD EXPIRED

To change your password - enter: nnn/nnn  where nnn = new password
or,
enter LOGOFF to cancel

 ICH70001I IBMUSER  LAST ACCESS AT **:**:** ON ****, **** **,****
 HCPRPW004I Password changed
RPIMGR031E RESOURCE MAINT.190 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE MAINT.19E SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 7VMRAC20.29E SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 7VMRAC20.505 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 7VMRAC20.191 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE RACFVM.305 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE IBMUSER.191 SPECIFIED BY LINK COMMAND NOT FOUND
z/VM Version 7 Release 2.0, Service Level 2001 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:24:34 EDT FRIDAY 20/09/22
z/VM V7.2.0    2020-06-26 09:03
...
```

4. Set the F12 function key to the command **RETRIEVE**:

```
===> set pf12 ret
```

5. Link and access 7VMRAC20's 505, 191, and 29E disks. Disregard any error messages:

```
===> link 7VMRAC20 505 505 rr
RPIMGR031E RESOURCE 7VMRAC20.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 0505 LINKED R/O; R/W BY RACMAINT
===> acc 505 c
DMSACP723I C (505) R/O
===> link 7VMRAC20 191 192 rr
RPIMGR031E RESOURCE 7VMRAC20.191 SPECIFIED BY LINK COMMAND NOT FOUND
===> acc 192 b
DMSACP723I B (192) R/O
DMSACP725I 192 also = D disk
===> link 7VMRAC20 29e 29e rr
RPIMGR031E RESOURCE 7VMRAC20.29E SPECIFIED BY LINK COMMAND NOT FOUND
===> acc 29e d
DMSACP724I 29E replaces D (192) R/O
DMSACP723I D (29E) R/O
```

6. Update the RACF database with existing CP directory information by using the **RPIBLDDS** command. The RPIDIRCT SYSUT1 file that was created earlier and copied to the 7VMRAC20 191 disk is used as input. You can again choose to issue the command **#CP TERM MORE 0 0** because many panels of messages will be issued:

```
===> rpibldds rpidirct
Processing batch file RPIDIRCT SYSUT1 using "RAC" command interface
...
 => PERMIT LOGONBY.SSLDCSSM CLASS(SURROGAT) ID(TCPMAINT) ACCESS(READ)
 => PERMIT LOGONBY.SSLDCSSM CLASS(SURROGAT) ID(GSKADMIN) ACCESS(READ)
 => setropts generic(vmbatch) gencmd(vmbatch)
 => rdefine vmbatch ** uacc(none)
 => permit ** class(vmbatch) id(ftpserve vmnfs dirmsat dirmsat2) acc(control)
 => setropts classact(vmbatch vmmdisk vmcmd vmlan surrogat)
```

The RACF database is now populated with the values from the user directory and other modifications that were configured previously.

7. Define the security administrator virtual machine. In this example, the default of SYSADMIN is used:

```
===> rac alu sysadmin special
```

8. Log off from IBMUSER.

9. Log on to SYSADMIN. You will be asked to change the password.

10. Grant OPERATIONS privileges to the following virtual machines:

```
===> rac alu datamove operations
===> rac alu MAINT720 operations
===> rac alu bldseg operations
===> rac alu lnxadmin operations
```

These commands give the four specified virtual machines access to all minidisks on the system.

11. Revoke the privileges for the IBMUSER virtual machine because it is no longer needed:

```
===> rac alu ibmuser revoke
```

12. Grant the DIRMAINT virtual machine SPECIAL privileges:

```
===> rac alu dirmaint special
```

13. Grant the MAINT virtual machine SPECIAL and OPERATIONS privileges:

```
===> rac alu maint special operations
```

14. Define the system virtual switches that are named VSW1 and VSW2 to the VMLAN class:

```
===> rac rdefine vmlan system.vsw1
===> rac rdefine vmlan system.vsw2
```

15. Permit TCPIP to the virtual switch VSW1:

```
===> rac permit system.vsw1 class(vmlan) id(tcpip) access(update)
```

16. Permit Linux machines to the virtual switch VSW1:

```
===> rac permit system.vsw1 class(vmlan) id(lnxadmin) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux1) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux2) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux3) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux4) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux5) access(update)
===> rac permit system.vsw1 class(vmlan) id(linux6) access(update)
```

17. Log off from SYSADMIN.

The initial RACF database is now configured.

Enabling RACF-connector on DirMaint on the first member

Note: For more information about enabling a RACF-connector, see 10.4, “DirMaint-RACF Connector” on page 321.

Complete the following steps to enable DirMaint to run to RACF:

1. Log on to MAINT. You are prompted to change the password:

```
===> logon maint
```

2. Link to the 7VMDIR20 2C2 disk read-only, which has a sample CONFIGRC DATADVH file:

```
===> vmlink 7VMDIR20 2c2
DMSVML2060I 7VMDIR20 2C2 linked as 0120 file mode Z
```

3. Copy the sample CONFIGRC file from the Z disk to the A disk as file type DATADVH:

```
===> copy configrc sampdvh z = datadvh a
```

4. Start DirMaint with the **XAUTOLOG DIRMAINT** command:

```
===> xautolog dirmaint
 ICH70001I DIRMAINT LAST ACCESS AT 15:38:05 ON WEDNESDAY, JUNE 20, 2012
Command accepted
Ready; T=0.01/0.01 15:50:02
AUTO LOGON  ***       DIRMAINT USERS = 5
HCPCLS6056I XAUTOLOG information for DIRMAINT: The IPL command is verified by
the IPL command processor.
 DVHPRO2008I ROLE = DIRMAINT
```

5. Add the CONFIGRC DATADVH configuration file to DirMaint with the **DIRM FILE** command. You can ignore error messages, such as the RPIMGR031E message that is shown:

```
===> dirm file configrc datadvh
```

```
RPIMGR031E RESOURCE DIRMAINT SPECIFIED BY SPOOL COMMAND NOT FOUND
RPIMGR031E RESOURCE POKDEV62 SPECIFIED BY TAG COMMAND NOT FOUND
PUN FILE 0011 SENT TO   DIRMAINT RDR AS  0004 RECS 0103 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
 DVHREQ2288I Your FILE request for MAINT at * has been accepted.
 DVHRCV3821I File CONFIGRC DATADVH A2 has been received; RC = 0.
 DVHREQ2289I Your FILE request for MAINT at * has completed; with RC = 0.
```

6. Issue the **DIRM RLDDATA** command so that the change is activated:

```
===> dirm rldd
DVHXMT1191I Your RLDDATA request has been sent for processing to
DVHXMT1191I DIRMAINT at POKDEV62.
DVHREQ2288I Your RLDDATA request for MAINT at * has been accepted.
 DVHITI6314E No DATAMOVE machines were defined in the config file.
 DVHREQ2289I Your RLDDATA request for MAINT at * has completed; with RC =
 DVHREQ2289I 0.
```

DirMaint is now initially enabled to use RACF-connector.

Setting the DirMaint use of the reader with RACF on the first member

Because the VMBATCH definitions were deleted in 6.7.1, “Creating the RACF RPIDIRCT command file” on page 156, RACF reports errors when DirMaint sends files to the reader. To address this issue, the CP **TRANSFER** and **TAG** commands must not be controlled.

In addition, SMAPI needs to issue commands for other users with the **FOR** command under privilege class C. To address this requirement, the CP **FOR.C** commands need to *not* be controlled.

To change these settings, complete the following steps:

1. Create a RACF profile for the VMXEVENT class named EVENT1:

```
===> rac rdefine vmxevent event1
```

2. Add three members to the VMEVENT class for the **TRANSFER** (privilege class G) command, the **TAG** command, and the **FOR** (privilege class C) command, and set them to no-control:

```
===> rac ralter vmxevent event1 addmem(transfer.g/noctl tag/noctl for.c/noctl)
```

3. Activate the VMXEVENT class:

```
===> rac setropts classact(vmxevent)
```

4. Refresh the VMEVENT class:

```
===> rac setevent refresh event1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: LINK
...
```

5. Log off from MAINT.

DirMaint and SMAPI are now enabled to run with RACF.

Putting RACF into production

RACF is now configured to be put into production. Put RACF into production with the following steps:

1. If you are OPERATOR, disconnect:

```
===> disc
...
```

2. Log on as **MAINT720** on the next member. You will be asked to change the password on the first member. On subsequent members, use the new password.
3. Start the AUTOLOG2 virtual machine with the **XAUTOLOG** command to start the shared file pool server machines:

```
===> xautolog autolog2
 ICH70001I AUTOLOG2 LAST ACCESS AT **:**:** ON ****, **** **,****
Command accepted
AUTO LOGON  ***       AUTOLOG1 USERS = 5
HCPCLS6056I XAUTOLOG information for AUTOLOG1: The IPL command is verified by
the IPL command processor.
```

Put RACF into production with the **PUT2PROD RACF** command. Watch for the "Completed successfully" message:

```
===> put2prod rac
...
```

4. Put CP into production with the **PUT2PROD CP** command. Watch for the completed successfully message:

```
===> put2prod cp
...                         // a number of screens pass by
VMFP2P2760I PUT2PROD processing completed successfully
```

RACF is now prepared to go into production at the next IPL.

5. Log off from MAINT720.
6. Log on to OPERATOR. You will be asked to change the password on the first member.
7. Log off the RACMAINT virtual machine with the **FORCE** command:

```
===> force racmaint
RACMAINT: CONNECT= 00:37:57 VIRTCPU= 000:03.32 TOTCPU= 000:04.03
RACMAINT: LOGOFF AT 16:11:53 EDT WEDNESDAY 06/20/12 BY OPERATOR
16:11:53 USER DSC   LOGOFF AS  RACMAINT USERS = 22    FORCED BY OPERATOR
16:11:53 HCPRPI036E CP/RACF communication path broken to RACMAINT
```

8. Start the RACFVM virtual machine with the **XAUTOLOG** command and watch for messages that indicate that RACF is starting:

```
===> xautolog racfvm
14:42:39 Command accepted
14:42:39 AUTO LOGON  ***       RACFVM   USERS = 23    BY OPERATOR
16:12:00 HCPCLS6056I XAUTOLOG information for RACFVM: The IPL command is
verifie
d by the IPL command processor.
RACFVM  : RACFVM CMS XA Rel 14 09/18/2020
RACFVM  : DMSACP723I B (305) R/O
RACFVM  : RACF is defined to the Z/VM system and the current product status is
ENABLED
RACFVM  :
RACFVM  :       RACF
RACFVM  :  Feature for z/VM
```

```
RACFVM  :  Version 7.2.0
RACFVM  :
RACFVM  :  Licensed Materials - Property of IBM
RACFVM  :  5741-A07
RACFVM  :  (C) Copyright IBM CORP. 1981, 2010  All Rights Reserved.
RACFVM  :
RACFVM  : DMSACC723I R (0200) R/W - OS
RACFVM  : DMSACC723I Q (0300) R/W - OS
...
16:12:02 HCPRPI035I CP/RACF communication path established to RACFVM
...
```

RACF is now running on the current member.

9. Shut down the member:

```
===> shutdown
...
00: 13:52:25 HCPWRP961W SYSTEM SHUTDOWN COMPLETE FOR LEFT630  ON 2012-06-22
00: HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000
00000961
```

For SSI members other than the first member, perform the steps in the first and last of the five subsections only:

- "IPLing the member and start RACMAINT" on page 167.
- "Putting RACF into production" on page 172.

After you perform the **PUT2PROD** sections on all SSI members, IPL the members one at a time from the default (*RES*) volume. Do not specify the NOAUTOLOG parameter. You will see RACF start on the OPERATOR console.

When the system comes back up, RACF is running.

6.7.7 Configuring SMAPI to work with RACF

In this section, we describe configuring SMAPI to work with RACF. You to perform these steps only if you use SMAPI.

For more information about SMAPI, see "Systems Management API" on page 324.

Complete the following steps to allow SMAPI to work with RACF:

1. Access your system through a 3270 emulator.
2. Log on to MAINT on the first SSI member.
3. Allow VSMWORK1 to have CONTROL authority of the z/VM minidisk (VMMDISK) that contains the SYSTEM CONFIG file (PMAINT CF0). Perform the following commands:

```
===> rac permit pmaint.cf0 class(vmmdisk) acc(control) id(vsmwork1)
===> rac permit maint.cf1 class(vmmdisk) acc(control) id(vsmwork1)
```

4. Allow VSMWORK1 to have CONTROL access to the generic class VMBATCH:

```
===> rac permit ** class(vmbatch) id(vsmwork1) access(control)
```

5. Allow SMAPI workers to read the `TCPMAINT 198` disk:

```
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmguard)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork1)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork2)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork3)
```

6. Allow `LNXADMIN` to read certain disks:

```
===> rac permit pmaint.cf0 class(vmmdisk) acc(read) id(lnxadmin)
===> rac permit autolog1.191 class(vmmdisk) acc(read) id(lnxadmin)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(lnxadmin)
```

7. Change the default password expiration to your security standard. We set ours to 90 days by using the following command:

```
===> rac setropts password(interval(90))
```

Enable RACROUTE

Enable the SMAPI service machines `VSMREQI6`, `VSMREQIN`, `VSMREQIU`, `VSMEVSRV`, `DTCSMAPI`, `VSMWORK1`, `VSMWORK2`, and `VSMWORK3` to use **RACROUTE** services with the following commands:

```
===> RAC SETROPTS CLASSACT(FACILITY)
===> RAC RDEFINE FACILITY ICHCONN UACC(NONE)
ICH10006I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE ADDITION(S)
UNTIL A SETROPTS REFRESH IS ISSUED.
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQI6) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIN) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIU) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMEVSRV) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(DTCSMAPI) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK1) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK2) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK3) ACCESS(UPDATE)
...
===> RAC SETROPTS RACLIST(FACILITY)
```

Exempting SMAPI from certain command checking

You need to make four SMAPI service machines (`DTCSMAPI`, `VSMWORK1`, `VSMWORK2`, and `VSMWORK3`) exempt from access checking. Even if access checking is not active on your system, make the SMAPI service machines exempt from access checking for the **FOR** (privilege class C) and **LINK** commands. Follow these steps:

1. Make the `DTCSMAPI` virtual machine exempt with the following commands:

```
===> RAC SETROPTS CLASSACT(VMXEVENT)
===> RAC RDEFINE VMXEVENT USERSEL.DTCSMAPI
===> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.DTCSMAPI
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
```

```
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

2. Make the VSMWORK1 virtual machine exempt with the following commands:

```
===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK1
===> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.VSMWORK1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

3. Make the VSMWORK2 virtual machine exempt with the following commands:

```
===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK2
===> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.VSMWORK2
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
```

```
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

4. Make the VSMWORK3 virtual machine exempt with the following commands:

```
===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK3
===> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.VSMWORK3
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

RACF can now allow SMAPI to do its job. It is recommended that you follow the instructions that are described in 3., “Test SMAPI from Linux by using smaclient.” on page 333 and 10.5.10, “Testing SMAPI from Linux by using smaclient” on page 339.

6.7.8 Configuring LogonBy processing

RACF can be configured to require users to log on with their own credentials. This procedure is called *LogonBy processing*. LogonBy processing is required for a correct *audit trail* because it allows SMF to capture each individual’s access. A similar procedure is also available for DirMaint in 6.14.2, “Using LOGONBY for correct accountability” on page 218.

The function of LOGONBY is similar to the use of SURROGAT class profiles in z/OS. It is a preferred practice that when a LOGONBY profile is defined for a generic virtual machine, it is no longer possible to use the standard password to log on.

The following example creates userid1 and gives it access to SYSADMIN:

1. Log on as MAINT.
2. Create a file that is called USERID1 DIRECT A with the following data:

```
===> x userid1 direct
USER USERID1 PASSWORD1 512M 1G G
```

3. Issue the **DIRM ADD** command for that virtual machine:

```
===> dirm add userid1
PUN FILE 0092 SENT TO   DIRMAINT RDR AS  0057 RECS 0011 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXMT1191I RDBKZVMG.
Ready; T=0.01/0.01 09:36:19
 DVHREQ2288I Your ADD request for USERID1 at * has been accepted.
 DVHBIU3450I The source for directory entry USERID1 has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHDRC3451I The next ONLINE will take place via delta object directory.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry USERID1 have been placed
 DVHBIU3428I online.
 DVHREQ2289I Your ADD request for USERID1 at * has completed; with RC
 DVHREQ2289I = 0.
 DVHREQ2288I Your DSATCTL request for DIRMAINT at
 DVHREQ2288I * has been accepted.
 DVHREQ2289I Your DSATCTL request for DIRMAINT at
 DVHREQ2289I * has completed; with RC = 0.
```

4. Set up the surrogate RACF class if it does not exist:

```
===> rac setr classact(surrogat)
===> rac setr generic(surrogat)
===> rac setr gencmd(surrogat)
===> rac setr classact(surrogat)
===> rac setr raclist(surrogat)
```

5. Allow logon by processing for `SYSADMIN` only:

```
===> rac rdef surrogat logonby.SYSADMIN audit(all)
```

6. Allow `SYSADMIN` to be logged on to by `USERID1`:

```
===> rac permit logonby.sysadmin cl(surr) acc(read) id(userid1)
===> rac setr raclist(surr) refresh
```

7. Test the log on, as shown in Figure 6-3.

```
    z/VM ONLINE

,.-,        +``-`_+.             .:(`  )`.  .--`++.,
    ,((    ).    _(   :  '`.          :(        -        )
  ,(        ).)(`(     :   )   . .--``` www.ibm.com/vm  -,.))+).
((    (..__.:'-' __.:'     .+.`(           :               +       )
.(      ) )   // VVVVV\ :        /VVV\MMMMMMMM             MMMMMMMM\
 -' __.:'    ,//  VVVVV\        /VVVV\M      M           M       M\
             //    VVVVV\      /VVVVV\M       M         M        M\
            ,//     VVVVV\    /VVVVV\ MMMMM    M       M     MMMMM\
            //       VVVVV\   /VVVVV\  \\\\M      M     M      M\\\\\
   ZZZZZZ\  ,//     (  VVVVV\ /VVVVV\       M   M   M   M   M   M\
   \\\ZZ\/  //   .+(`(  VVVVV\VVVVV\        M   MM   M M   MM   M\
     ZZ\/  ,//  (       VVVVVVVVV\          M   M\M   M   M\M   M\
    ZZ\/   //  ( (       VVVVVVV\       MMMMM   M\ M     M\ M   MMMMM\
   ZZ\/   ,//   ( .       VVVVV\        M       M\  M   M\  m       M\
  ZZZZZZ\ //     .  _      VVV\         M       M\ : M M\   M       M\
  \\\\\\/,//     (          V\        , MMMMMMMMM\    M\    MMMMMMMMM\
                   `___    .. \  __..:   \\\\\\\\\          \\\\\\\\\\
             LL                                 DD
  ,ccCCCCc,  LL  ,oOOOOOo,  UU        UU  ,ddDDDd,DD    Built on IBM
 :cC"        LL :O"     "O' UU        UU dD"    `DDD    Virtualization
 ."Cc,   ,c  LL :Oo,   ,oO' "Uu,   ,uUU "Dd,   ,dDD    Technology
  `"CccccC"  LLL `"OOoOO"    `"UUUUU'UU  `"DdddD"DDD

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID   ===>
PASSWORD ===>

COMMAND  ===>  logon sysadmin by edialves
                                                          RUNNING
RDBKZVMG
```

Figure 6-3 Test the logon

You are prompted to change the password at the first log on:

```
logon sysadmin by edialves

Enter your password,
or
To change your password, enter: ccc/nnn/nnn
     where ccc = current password, and nnn = new password

RPIMGR042I PASSWORD EXPIRED

To change your password - enter: nnn/nnn  where nnn = new password
or,
enter LOGOFF to cancel

 ICH70001I SYSADMIN LAST ACCESS AT 09:58:11 ON TUESDAY, JUNE 11, 2013
```

```
 HCPRPW004I Password changed
z/VM Version 7 Release 2.0, Service Level 2001 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:10:58 EDT TUESDAY 06/11/13
z/VM V7.2.0    2020-06-04 12:50
```

You can issue a **QUERY USERID** command to see that you are logged on as SYSADMIN with its privileges.

6.7.9 Using the RACF SMF data unload utility

The *RACF SMF data unload utility* is a simple way to extract RACF type 80 SMF data. The following example shows the TYPE 80 SMF record where USERID1 was created and given access to log on by SYSADMIN. The virtual machine that will access the RACFADU EXEC will need RACF AUDITOR access. It will need to link to the RACFVM SMF output disks 301 and 302. The utility **RACFADU** is on the RACFVM 305 disk.

Complete the following steps:

1. Log on to MAINT.
2. Link the RACF 301, 302, and 305 disks:

```
===> link racfvm 301 301
===> link racfvm 302 302
===> link racfvm 305 305
===> acc 305 b
```

> **Note:** To access the RACFVM 301 disk, you need RACF AUDITOR privileges.

3. Run the **RACFADU EXEC** by using the 301 disk as input and the 191 disk as output.

> **Note:** The RACFADU will work only if the output disk (191) is accessed as filemode A. In this example, the output file is twice the size of your 301 used space.

```
===> RACFADU 301 191
RACFADU  OUTPUT
  RPIADU033I SMF unload completed successfully.
  View the RACFADU MESSAGES file for additional details.
```

The output is now in the RACFADU OUTPUT A file.

6.8 Implementing more network features

The following recommended changes to the system are described:

- Enable z/VM FTP and Network File System functionality
- Reconfigure TCP/IP for high availability by using a VSWITCH

The main TCP/IP configuration file is the `PROFILE TCPIP` file and it is on the `TCPMAINT 198` disk, which is accessed as the `D` disk.

6.8.1 Enabling z/VM FTP and Network File System functions

Enable both the FTP and Network File System (NFS) functions by completing the following steps:

1. Log on to **`TCPMAINT`** by using the password that you set for service machines in section 10.2.1, “Changing default passwords” on page 310.
2. Make a backup copy of the TCP/IP configuration file, `PROFILE TCPIP D`:

   ```
   ===> copy profile tcpip d = tcpiorig = (olddate
   ```
3. Edit the TCP/IP configuration file:

   ```
   ===> xedit profile tcpip d
   ```
4. Make the following changes:
   a. Locate the last line that begins with **`OBEY`** and move your cursor into the prefix area beside the line underneath that begins with **`OPERATOR`**.
   b. Enter **`I`** in the prefix area and press Enter to add a single blank line.
   c. Enter **`LGLOPR WAVEWRKS WAVEWRKL`**, as shown in Example 6-12 on page 181,
   d. Locate the last line that begins with “**`;” 2049`** and move your cursor into the prefix area on the next line that begins with “**`; -----`** “
   e. Enter **`I3`** into the prefix area and press Enter to add three blank lines.
   f. As shown in Example 6-12 on page 181, enter an **`AUTOLOG`** statement, add **`FTPSERVE X`** to the next line for logging onto the FTP server when TCP/IP starts, and **`ENDAUTOLOG`** as the last statement.
   g. In the `PORT` stanza, remove the semicolons to uncomment the lines with `FTPSERVE` on them (ports 20 and 21) and the lines with `VMNFS` (port 2049 TCP and UDP).

   These changes will cause FTP and NFS services to start when TCP/IP is started.

   The important lines before the file is edited are shown in Example 6-11.

Example 6-11 Initial PROFILE TCPIP file

```
...
; ------------------------------------------------------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE REXECD SNMPD SNMPQE LDAPSRV MAINT720
ENDOBEY
; ------------------------------------------------------------------------
PORT
; 20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
; 21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
...
```

```
; 2049 UDP VMNFS              ; NFS Server
; 2049 TCP VMNFS     NOAUTOLOG ; NFS Server
; ---------------------------------------------------------------------
...
```

The lines after the file is edited are shown in Example 6-12.

Example 6-12 Modified PROFILE TCPIP file

```
...
; ---------------------------------------------------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE REXECD SNMPD SNMPQE LDAPSRV MAINT720
LGLOPR WAVEWRKS WAVEWRKL
ENDOBEY
; ---------------------------------------------------------------------
PORT
   20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
   21   TCP FTPSERVE            ; FTP Server
   23   TCP INTCLIEN            ; TELNET Server
...
   2049 UDP VMNFS               ; NFS Server
   2049 TCP VMNFS     NOAUTOLOG ; NFS Server
; ---------------------------------------------------------------------
AUTOLOG
  FTPSERVE X
ENDAUTOLOG
; ---------------------------------------------------------------------
...
```

5. Save your changes with the **FILE** subcommand:

   `====> file`

6. Repeat these steps on all other members of the SSI cluster.

6.8.2 Reconfiguring TCP/IP for high availability by using a VSWITCH

The previous configuration of the TCP/IP virtual machine was only an initial configuration that was intended to activate the network stack as quickly as possible.

Characteristics of VSWITCH interfaces

VSWITCH interfaces have the following characteristics:

- VSWITCHES are run by a set of redundant virtual service machines, by default.
- VSWITCHES are able to fail over with up to three real devices.
- VSWITCHES can be configured to be VLAN aware.
- Up to 2,048 virtual network interfaces can be coupled to a single VSWITCH.
- Ports on VSWITCHES can be configured either USER based or with port numbers.
- Both access and trunk ports can be configured for VSWITCHES.
- VSWITCH network interfaces always operate on port 0 of the virtual device.

The VSWITCH that was defined earlier as `VSW1` in the system configuration file has two different connections to the network. It is considered highly available. You must now modify the TCP/IP configuration to use the virtual switch so that its OSA devices are not a single point of failure.

Modify TCPIP parm files

Complete the following steps:

1. On node 1, log on as **TCPMAINT** by using the password that you set for service machines.
2. Edit the SYSTEM DTCPARMS file on the TCPMAINT 198 (D) disk.
3. Comment out the last line by inserting a period followed by an asterisk (.*) in the first two columns, which will prevent the OSA triplet from being directly attached to the TCP/IP virtual machine on start-up:

```
===> xedit system dtcparms D
.*******************************************************
.* SYSTEM DTCPARMS created by DTCIPWIZ EXEC on 8 Apr 2015
.* Configuration program run by MAINT720 at 15:37:40
.*******************************************************
:nick.TCPIP    :type.server
               :class.stack
.*             :attach.2103-2105

====> file
```

4. Make a backup copy of the working PROFILE TCPIP file that was created by the **IPWIZARD**:

```
===> copy profile tcpip d = tcpiwrks = (olddate
```

5. Edit the PROFILE TCPIP file on the TCPMAINT 198 (D) disk. Change the real OSA starting address (***2103*** in this example) to the virtual starting address (0600) everywhere in the file:

```
===> xedit profile tcpip d
====> c/2103/0600/* *
DMSXCG517I 4 occurrence(s) changed on 3 line(s)
====> file
```

   This command instructs TCPIP to use the virtual NIC that starts at the virtual device address 600.

6. Log off from TCPMAINT.
7. **Repeat these steps** on all other members in the cluster. Remember, the real OSA addresses might differ on each node.

Modifying the TCPIP user directory entry

> **Note:** Commands that modify directory entries are processed exactly as they are entered. So, for consistency, it is a preferred practice to enter DirMaint commands in *all uppercase characters*, although it is not required.

Complete the following steps:

1. Log on to the first node of the SSI cluster as MAINT.
2. Create a virtual NIC for the TCP/IP virtual machine on VSW1:

```
===> DIRMAINT FOR TCPIP NICDEF 0600 TYPE QDIO LAN SYSTEM VSW1
```

3. Ask DirMaint for the list of CP COMMAND statements that are defined for TCPIP. At this point, it states that no COMMAND statements exist:

```
===> DIRMAINT FOR TCPIP COMMAND ?
DVHXMT1191I Your COMMAND request has been sent for processing ...
Ready; T=0.01/0.01 23:34:49
 DVHREQ2288I Your COMMAND request for TCPIP at * has been accepted.
   DVHCOM3581I There are no COMMAND statements to query.
```

```
DVHREQ2289I Your COMMAND request for TCPIP at * has completed; with RC = 0.
```

4. Add a command to the TCP/IP directory definition to allow it to access the VSWITCH:

   ```
   ===> DIRMAINT FOR TCPIP COMMAND ADD 001 SET VSWITCH VSW1 GRANT &USERID
   ```

   These statements grant TCP/IP access to VSWITCH `VSW1`, define a virtual NIC that starts at virtual device address `600`, and couple it to the VSWITCH.

   > **Note:** If RACF is enabled on your system, invoke the following commands:
   >
   > ```
   > RAC PERMIT SYSTEM.vsw1 CLASS(VMLAN) ID(tcpip) ACCESS(UPDATE)
   > RAC SETROPTS CLASSACT(VMLAN)
   > ```

The z/VM TCP/IP stack comes up on the highly available VSWITCH the next time that you IPL z/VM.

6.9 Shutting down and IPLing the SSI cluster again

You can watch the z/VM member shut down and IPL again from the Integrated 3270 Console. If you issue this command from a 3270 emulator, you will lose your session and will not see most of the shutdown process. To shut down and IPL the SSI cluster again, perform the following steps:

1. Log off from `MAINT` and `MAINT720` on all 3270 emulator sessions.
2. Start an Integrated 3270 Console session for the LPAR of the first SSI cluster member.
3. Log on to `MAINT`.
4. Using the **AT** command, issue the **SHUTDOWN** command for all other members. In this example, the system name is ***RDBKZVMH***:

   ```
   ===> at RDBKZVMH cmd shutdown
   HCPSHU960I System shutdown may be delayed for up to 560 seconds
   ...
   ```

   An informational message displays that indicates the possible time delay while z/VM waits for virtual machines to quiesce and log off. In this example, 560 seconds is the sum of the two shutdown values that were set in the SYSTEM CONFIG file during 6.5.1, “Modifying features and optimizing parameter settings” on page 144.

   If more than two members exist, repeat the **AT *NODE* CMD SHUTDOWN** step for those members.

5. After the other nodes successfully complete their shutdowns, they turn red in the HMC with a status of `NOT RUNNING`. When all other nodes are down, issue the **SHUTDOWN REIPL** command to node 1:

   ```
   ===> shutdown reipl
   ...
   ```

   All members of the SSI cluster are now down, and member 1 is coming back up. z/VM typically will IPL extremely fast, usually in less than a minute.

6. When z/VM comes back up, you see messages while the system IPLs, and finally the z/VM logon panel.
7. Try to start a 3270 emulator session to member 1 by using the DNS host name or assigned IP address. You see the z/VM logon panel. If not, you must debug the problem from the Integrated 3270 Console session. For example, you can execute `FORCE TCPIP` and log on to TCP/IP interactively and watch for error messages.

8. Verify that TCP/IP is attached with the **QUERY VSWITCH** with **DETAILS** command:

```
===> query vswitch vsw1 details
VSWITCH SYSTEM VSW1     Type: QDIO    Connected: 1    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0A-00-00-01    MAC Protection: Unspecified
  IPTimeout: 5        QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
Uplink Port:
  State: Ready
  PMTUD setting: EXTERNAL   PMTUD value: 8992
  RDEV: 2100.P00 VDEV: 0600 Controller: DTCVSW2  ACTIVE
        EQID: OSA1SET1
    Uplink Port Connection:
      RX Packets: 45        Discarded: 0         Errors: 0
      TX Packets: 82        Discarded: 0         Errors: 0
      RX Bytes: 3330                TX Bytes: 12478
      Device: 0600  Unit: 000   Role: DATA      Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
  RDEV: 2120.P00 VDEV: 0600 Controller: DTCVSW1  BACKUP
        EQID: OSA1SET1
 Adapter Connections:
    Adapter Owner: TCPIP    NIC: 0600.P00 Name: UNASSIGNED  Type: QDIO
      RX Packets: 5044      Discarded: 0         Errors: 0
      TX Packets: 82        Discarded: 0         Errors: 0
      RX Bytes: 220405              TX Bytes: 12478
      Device: 0600  Unit: 000   Role: DATA      Port: 0003
      Options: Ethernet Broadcast
        Unicast MAC Addresses:
          02-00-0E-0A-00-05 IP: 9.60.87.13
        Multicast MAC Addresses:
          01-00-5E-00-00-01
```

Member 1 is back up with TCP/IP attached to the highly available VSWITCH and the FTP server is running.

6.9.1 IPLing the other SSI members

Complete the following steps to IPL the other SSI members:

1. Go to the HMC and start an Integrated 3270 Console for the second SSI member.
2. IPL the LPAR with the **Load** task.
3. Go to the Integrated 3270 Console and complete the IPL of z/VM:
   a. Press F10 at the SAPL window.
   b. Type **WARM** at the **Start** command.
   c. Type NO at the request to reset the Time of Day (TOD) clock.
   d. If you are prompted for anything that requires a response of **go**, type GO.
4. Disconnect from OPERATOR on the Integrated 3270 Console by typing **DISCO HOLD**. You will see a z/VM logon panel.
5. If more than two member nodes are in your cluster, repeat steps 1 through 4 to start those members.

6. Verify that the other nodes in the cluster can be accessed through the highly available VSWITCH.

The entire SSI cluster is now back up.

6.10 Validating and testing your changes

To test several of the changes that you made, perform the following steps:

1. Start a 3270 emulator session to the first SSI member.
2. Log on as `MAINT`.
3. Issue the **`QUERY SSI`** command:

```
===> query ssi
SSI Name: ITSOSSIA
SSI Mode: Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: VV155A on 155A
SLOT SYSTEMID STATE     PDR HEARTBEAT       RECEIVED HEARTBEAT
   1 RDBKZVMG Joined    2015-04-26 17:53:11 2015-04-26 17:53:11
   2 RDBKZVMH Joined    2015-04-26 17:53:04 2015-04-26 17:53:04
   3 -------- Available
   4 -------- Available
```

4. Use the **`QUERY RETRIEVE`**, **`QUERY VDISK`**, and **`SSICMD QUERY VDISK`** commands to see the changes that were made to the `Features` statement in the `SYSTEM CONFIG` file:

```
===> query retrieve
99 buffers available.  Maximum of 255 buffers may be selected.
===> query vdisk userlim
VDISK USER   LIMIT IS    2097152 BLK
===> ssicmd query vdisk userlim
RDBKZVMG:
VDISK USER   LIMIT IS    2097152 BLK

RDBKZVMH:
VDISK USER   LIMIT IS    2097152 BLK
===> ssicmd query vdisk syslim
RDBKZVMG:
VDISK SYSTEM LIMIT IS INFINITE,          0000000 BLK IN USE

RDBKZVMH:
VDISK SYSTEM LIMIT IS INFINITE,          0000000 BLK IN USE
```

5. Try to start an FTP session to all of the SSI members. You will get a logon prompt.

This test shows that the changes to the `SYSTEM CONFIG` file and to the FTP server are in effect.

6.11 Adding page volumes and perm (user) volumes

Each z/VM 6.3 SSI member is installed with one paging volume and one spool volume, either 3390-3s or 3390-9s, depending on which type of disks the cluster was installed onto.

One spool volume for each member is probably adequate for Linux needs. However, more paging volumes are recommended, especially if you plan to use the z/VM memory overcommitment feature for your Linux virtual machines.

Although certain volumes are shared, the page and temporary disk volumes are not.

If you used 3390-9, it is recommended that you add at least one additional 3390-9 paging volume so that you will have a total of two. If you used 3390-3, add at least four additional 3390-3 paging volumes for a total of five. Adequate paging space will give you room to add more Linux virtual machines. Guidelines for planning paging were covered in 2.7, “Paging” on page 50.

6.11.1 Formatting volumes for page space

Before you add paging volumes to the SSI cluster members, you must format the DASD volumes to be used for minidisk space (`PERM`) and paging space (`PAGE`). Normally, you format the DASD volumes one at a time by using the **CPFMTXA** command.

If only a few volumes are involved, that is fine, but when you must format many volumes, the process of running **CPFMTXA** can be time-consuming and tedious, which can lead to errors. Therefore, a REXX EXEC that is named **CPFORMAT** is provided in the tar file that is associated with this book. With it, you can format many volumes with a single command. This EXEC is shown in Appendix B, “Reference, cheat sheets, blank worksheets, and education” on page 475. It is a wrapper around **CPFMTXA**. To use this EXEC, each DASD to be formatted must first be attached with the virtual device address and the same real device address (by using **ATTACH *realDev* ***).

This EXEC will label the volumes according to the convention that is described in 2.11.1, “DASD volume labeling convention” on page 58. If you want different volume labels, you can use the **CPFMTXA** command and manually specify each volume label, or you can modify the REXX EXEC. The use of **CPFMTXA** manually is covered in Chapter 15, “Miscellaneous recipes and helpful information” on page 439.

If you plan to install a systems management product, be aware of any volume labeling requirements that you must consider, such as the inclusion of the real device address.

6.11.2 Copying the utilities to Shared File System file pools

Complete the following steps:

1. Log on as **MAINT720** on the first member from your terminal emulator.
2. Issue the following command to create a new directory underneath the `MAINT720` ID in the system-generated clustered product Shared File System (SFS) pool, `VMPSFS`:

   ```
   ===> CREATE DIRECTORY VMPSFS:MAINT720.UTILS
   ```

   > **Note:** We use the VMPSFS filepool as a staging area because VMPSFS is created at the VMSSI cluster level. We move items out of this filepool into either the VMSYSU filepool on each member, or into the LNX filepool that we create later in this chapter. Consider VMPSFS to be volatile space, subject to overlay during system service.

3. Issue the following commands to create new directories under the MAINT ID on the userspace filepool, VMSYSU:

```
===> CREATE DIRECTORY VMSYSU:MAINT.UTILS
===> CREATE DIRECTORY VMSYSU:MAINT.UTILS.VMARC
```

4. Use VMLINK to access the TCP/IP tools so that you can use the z/VM FTP client:

```
===> vmlink tcpmaint 592
DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
```

5. Access the new SFS directory as file mode **M** with read/write mode:

```
===> access VMPSFS:MAINT720.UTILS. M (forcerw
```

6. On the FTP server. the directory path /ftp/zvm/cookbook/maintvrm (or /ftp/zvm/sg248147/maintvrm) was created automatically through the expansion of the .tgz file in 5.2.1, "Creating directories on the FTP server and upload the installation image" on page 108. We are now ready to transfer files from the FTP server to the VMPSFS file pool using one of the following methods:

   – Perform a **get** (pull) from z/VM by initiating a session to the FTP server:

```
===> ftp 9.60.87.87
...
===> lcd M
Local directory mode is 'M'
Command:
===> cd /ftp/zvm/cookbook/maintvrm
===> mget *.EXEC
===> mget *.PROFEXEC
===> mget *.AUTHFOR
===> mget *.CONTROL
===> mget *.XEDIT
===> mode s
>>>MODE s
200 S OK
Command:
===> binary fixed 1024
>>>TYPE i
200 TYPE is now 8-bit binary
Command:
===> mget *.VMARC
===> mget *.MODULE
===> quit
>>QUIT
221-Goodbye...
```

   – Perform a **put** (push) from the FTP server by completing the following steps:

      i. Issue the following command while logged on as **MAINT720** to any member of the z/VM cluster:

```
===> enroll administrator ftpserve vmpsfs
```

      The use of this command temporarily grants administrative access to this SFS pool for the virtual machine that is running the z/VM FTP server. Without this permission, you cannot issue the **cd** command from an FTP session.

      ii. Log on to your FTP server and start an FTP session to z/VM:

```
$ ftp RDBKZVMG.itso.ibm.com
Connected to RDBKZVMG.itso.ibm.com.
```

```
220-FTPSERVE IBM VM Level 630 at ATSVME6.ENDICOTT.IBM.COM, 21:10:02 ...
220 Connection will close if idle for more than 5 minutes.
Name (RDBKZVMG.itso.ibm.com:pwnovak):
MAINT720
331 Send password please.
Password:
230 MAINT720 logged in; working directory = MAINT720 191
Remote system type is z/VM.
cd VMPSFS:MAINT720.UTILS
ftp> cd VMPSFS:MAINT720.UTILS
```

> **Note:** At this point, if you skipped the administration enrollment process from the previous step, you see the following message:
>
> ```
> 550 FTP server does not have administrator authority for this file
> pool;  directory remains MAINT720.191
> ```

```
ftp>
```

```
lcd /ftp/zvm/cookbook/maintvrm

At this point, you may see the following:
ftp 9.60.86.30
Connected to 9.60.86.30.
220-FTPSERVE IBM VM Level 630 at ATSVME6.ENDICOTT.IBM.COM, 21:10:02 EDT
THURSDAY 2016-06-16
220 Connection will close if idle for more than 5 minutes.
Name (9.60.86.30:pwnovak): MAINT720
331 Send password please.
Password:
230 MAINT720 logged in; working directory = MAINT720 191
Remote system type is z/VM.
ftp> cd VMPSFS:MAINT720.UTILS

===>
===> mput *
===> quit
```

7. Check the listing of the files that you just downloaded into SFS:

```
===> vmfclear # listfile * * M (isodate
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS      DATE     TIME
CALLSM1  EXEC     M1 V         75        853          8 2015-04-28 17:43:26
CPFORMAT EXEC     M1 V         77        272          3 2015-04-28 17:43:26
SSICMD   EXEC     M1 V         64         71          1 2015-04-28 17:43:26
VMWW2    VMARC    M1 V       8192        106        203 2015-04-28 17:32:29
VMLOGS   VMARC    M1 V       8192          2          4 2015-04-28 17:32:28
VMSERVE  VMARC    M1 V       8192         28         52 2015-04-28 17:32:28
VMARC    MODULE   M1 V       8192          2          4 2015-04-28 17:32:27
VMCRON   EXEC     M1 V       1165          1          1 2015-04-28 17:33:32
```

8. Move the VMARC items into the VMARC directory:

```
===> access VMPSFS:MAINT720.UTILS.VMARC P (forcerw
===> copy * VMARC M = = P (OLDDATE
===> copy VMARC MODULE M = = P (OLDDATE
===> access VMSYSU:MAINT.UTILS.VMARC Q(forcerw
===> copy * VMARC M = = Q (OLDDATE
```

```
===> copy VMARC MODULE M = = Q (OLDDATE
===> erase * VMARC M
===> erase VMARC MODULE M
```

9. Deblock the VMARC module by using this pipeline so that it can be used later:

```
===> PIPE < VMARC MODULE P | deblock cms | > VMARC MODULE P
```

10. Create an enrollment for VMWW2 in the VM Product SFS (VMPSFS) file pool, access the directory, and move VMWW2 VMARC into that directory:

```
===> enroll user vmww2 vmpsfs ( blocks 5000
===> access vmpsfs:vmww2 z (forcerw
===> copy VMWW2 VMARC M = = Z
===> erase VMWW2 VMARC M
```

11. Release the VMARC directory (P) and the VMWW2 directory (Z):

```
===> release P
===> release W
```

12. You do not need to repeat these steps because the VMPSFS file pool is shared across all member nodes.

> **Important:** The VM Product SFS file pool (VMPSFS) is used to hold IBM product service data. It is a global (clustered) file pool, which is identified to CP as a global resource for SFS and z/VM Byte File System (BFS) functions. The items that are added to VMPSFS are a few small tools and utilities that will be accessed exclusively by z/VM system programmers and administrators.
>
> Do not add user data into this file pool. Instead, use VMSYSU, which is intended to hold user data.
>
> If you choose to add additional content to this SFS directory in the future, you must use caution to ensure that you do not fill up the only data storage group, group 2, beyond around 80%. You can check utilization with the **TALLY** command on the MAINT 193 minidisk. If the disk is not already accessed, use VMLINK:
>
> ```
> ===> VMLINK MAINT 193 ( INVOKE TALLY VMPSFS
> ```

13. Update the PROFILE EXEC for both MAINT and MAINT720 by adding the following line underneath the last ACCESS entry, which will cause this SFS directory to be accessed as file mode M at logon:

```
'ACCESS 51D D'
'ACCESS 551 E'
'EXEC VMLINK .DIR VMPSFS:MAINT720.UTILS. < . M * > (NON'
'SET FILEPOOL ...
```

Each SSI member now has access to the **CALLSM1**, **CPFORMAT**, and **SSICMD EXEC**s.

6.11.3 Using the CPFORMAT EXEC

To use the **CPFORMAT** EXEC, complete the following steps:

1. Log in to **MAINT** on the first member.
2. Use the **FILELIST** command to list the files on the SFS directory that are accessed as file mode M (which were configured in 6.7, "Enabling and configuring RACF" on page 155):

```
===> filelist
MAINT FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
Directory = VMPSFS:MAINT720.UTILS
Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
    CALLSM1  EXEC     M1 V         75        853          8 2015-04-28 17:43:26
    CPFORMAT EXEC     M1 V         77        272          3 2015-04-28 17:43:26
    SSICMD   EXEC     M1 V         64         71          1 2015-04-28 17:43:26
```

3. Move your cursor to the line with the CPFORMAT EXEC on it, and then either type an `X` (to indicate that you want to use XEDIT on that file) or press PF11 to invoke XEDIT for the file. Edit the file to set the first character that will be used in labels. Look for the variable **firstChar**:

```
===> xedit cpformat exec
====> /firstChar
/*************************************************************/
...
  Address COMMAND
  firstChar = 'V'
...
```

   If you want the first character in the labels to be a letter other than `V`, change this setting.

4. You can get brief help on **CPFORMAT** by using a question mark (**?**) parameter:

```
===> cpformat ?

Synopsis:

  Format one or a range of DASD as page, perm, spool or temp disk space
  The label written to each DASD is J<t><xxxx> where:
    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)
    <xxxx> is the 4 digit address

Syntax is:
                <----------------<
   >>--CPFORMAT--.-vdev--------.--AS---.-PERM-.---------><
                 '-vdev1-vdev2-'       '-PAGE-'
                                       '-SPOL-'
                                       '-TEMP-'
```

   The following example shows how to attach a 3390-9 volume and use **CPFORMAT** to format it as paging space. Refer to the planning worksheets that must be completed in Table B-9 on page 486. Our values are shown in Table 2-10 on page 67.

> **Important:** Because these volumes are formatted as page, the CPFORMAT EXEC will also add owner information to the DASD. For this reason, page volumes must be formatted on the SSI member on which they will be used.

5. The DASD that is used for the second paging volume on member 1 in this example is at real device address ***1565***. Query the device to see the status:

```
===> query 1565
DASD 1565 NW1565
```

6. Attach the device to MAINT by using the **ATTACH** command. This example uses the last parameter of *, which means the current virtual machine:

```
===> attach 1565 to *
DASD 1565 ATTACHED TO MAINT 1565 WITH DEVCTL HYPERPAV BASE
```

7. Use the **CPFORMAT** command with the **AS PAGE** parameter:

```
===> cpformat 1565 AS PAGE
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1565 MAINT    1565 3390 NW1565 1565          0      10017

WARNING - this will destroy data!
Are you sure you want to format the DASD as PAGE space (y/n)? y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1565 MAINT    1565 3390 VP1565 1565          0      10017
```

This formatting job might run for several minutes, depending on many factors.

8. Repeat the three previous steps on all other SSI members. In the example environment that was used for this book, two more page volumes were added on the second z/VM system in the cluster, RDBKZVMH.

6.11.4 Formatting DASD for minidisks

In addition to CP disks, such as page space, you will need system disks to create minidisks for the virtual machines. In the following steps, the DASD that will be used for virtual machine minidisks will be formatted. Perform these steps:

1. Start a 3270 session as MAINT on the first SSI cluster member.
2. Query the DASD that will be used for minidisks. In this example, the DASDs have real device addresses ***1567-1569 156A-156F***:

```
===> query 1567-1569 156A-156F
DASD 1567 NW1567 , DASD 1568 NW1568 , DASD 1569 NW1569 , DASD 156A NW156A
...
```

3. Attach the volumes:

```
===> attach 1567-1569 156A-156F to *
DASD 1567 ATTACHED TO MAINT 1567 WITH DEVCTL HYPERPAV BASE
...
```

4. Run the **CPFORMAT** command against these volumes and use the **AS PERM** parameter:

```
===> cpformat 1567-1569 156A-156F AS PERM
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1567 MAINT    1567 3390  NW1567 1567          0      10017
...
WARNING - this will destroy data!
Are you sure you want to format the DASD as PAGE space (y/n)? y
```

```
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1567 MAINT    1567 3390  VM1567 1567          0      10017
...
MAINT    156F MAINT    156F 3390  VM156F 156F          0      10017
```

Now, many volumes can be used for minidisks. The labels are prefixed with **VM** in this example.

6.11.5 Updating the SYSTEM CONFIG file

Now that the PAGE and PERM volumes are ready for use, they must be added to the SYSTEM CONFIG file. Follow these steps to update the SYSTEM CONFIG file:

To make these changes, complete the following steps as MAINT720:

1. Use VMLINK to access the PMAINT CF0 disk as multi-read/write (**MR**) and file mode **F**:

```
===> vmlink pmaint CF0 < * F MR >
DMSVML2060I PMAINT CF0 linked MR as 0120 file mode F
```

2. Rename the previous backup. Then, make a new backup copy of the previously edited SYSTEM CONFIG file by using the **COPYFILE** command with the **OLDDATE** parameter so that the time stamp of the file is not modified. Because the target file name (SYSTEM) and mode (F) are the same, the equal sign (=) can be used to indicate that the value from the source file needs to be reused for the target:

```
===> copy system conforig f -1system conforig = (olddate
===> copy system config f = conforig = (olddate replace
```

3. Check to ensure that your backups are present:

```
===> listfile sys* conf* F (ISO
FILENAME FILETYPE FM FORMAT LRECL  RECS  BLOCKS   DATE        TIME
SYSTEM   CONFIG   F1 V      80     378   5        2020-09-19  09:56:31
SYSTEM   CONFORIG F1 V      80     378   5        2020-09-19  09:56:31
-1SYSTEM CONFORIG F1 V      80     378   5        2020-09-19  09:48:05
```

   Ordinarily, we make a new copy of the SYSTEM CONFIG file by using the “WRKS” (it works) suffix convention, but because we did not IPL again yet, we do not know that the last edited version is correct yet. The -1 that is added to the beginning of the file name is used to indicate that it is the current version minus one.

4. Edit the SYSTEM CONFIG file and specify each of the new page volumes (PAGE) by name as CP_Owned. When your system IPLs, it will pick up these volumes as paging volumes:

```
===> xedit system config f
====> /page and
```

The pertinent information *before* the modification is shown in Figure 6-4.

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

    RDBKZVMG:   BEGIN
                CP_Owned   Slot 255  VP155E
    RDBKZVMG:   END

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

    RDBKZVMH:   BEGIN
                CP_Owned   Slot 255  VP1562
    RDBKZVMH:   END
```

Figure 6-4 SYSTEM CONFIG file before modification

The pertinent information *after* the modification is shown in Figure 6-5.

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

    RDBKZVMG:   BEGIN
                CP_Owned   Slot 253  VP155E
                CP_Owned   Slot 254  VP1565
    RDBKZVMG:   END

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

    RDBKZVMH:   BEGIN
                CP_Owned   Slot 253  VP1562
                CP_Owned   Slot 254  VP1566
    RDBKZVMH:   END
```

Figure 6-5 SYSTEM CONFIG file after modification

5. Move down to the `User_Volume_List` section. User volumes (`PERM`) can be specified individually with the `User_Volume_List` statement, or with wildcards by using the `User_Volume_Include` statement. If you are using the labeling convention that is enforced by the **`CPFORMAT EXEC`** and no other LPAR will use the same volumes with the same prefix, you can use wildcards with the `User_Volume_Include` statement. In Example 6-13, all volume labels that begin with ***`VM15`*** will be attached to `SYSTEM` and be available for the creation of minidisks.

Example 6-13 Adding volumes to the system configuration file

```
====> /user_v
/*                          User_Volume_List                          */
/**********************************************************************/
/* These volumes contain the minidisks for your guests, as well as    */
/* the product disks for each installed release of z/VM in the SSI    */
/* cluster.  Volumes that hold "local" minidisks, i.e., minidisks     */
/* unique to a single member system, should be wrapped in BEGIN/END   */
```

```
/* statement.  If it becomes necessary to access a local minidisk     */
/* from a different member of the SSI cluster operating in REPAIR     */
/* mode, simply ATTACH the volume to SYSTEM.                          */
/*********************************************************************/

/*********************************************************************/
/* Shared User Volumes                                                */
/*********************************************************************/
              User_Volume_List    VM155F
              User_Volume_Include VM5*
...
====> file
```

> **Important:** If other z/VM LPARs might attach volumes with the VM prefix, specifically list each volume to attach to `SYSTEM` by using the `User_Volume_List` statement. This step will prevent multiple z/VM systems from writing to the same volume. The list looks like the following list in this example:
>
> ```
> User_Volume_List VM1567
> User_Volume_List VM1568
> User_Volume_List VM1569
> User_Volume_List VM156A
> User_Volume_List VM156B
> ...
> ```
>
> This specification is another reason to correctly configure the Input/Output Definition File (IODEF) for each LPAR so that only DASDs that are pertinent to that LPAR are visible. Separations and fencing are good.

6. Save your changes with the **FILE** subcommand. Verify the integrity of the changes with the **CPSYNTAX** command:

   ```
   ===> access 193 g
   ===> cpsyntax system config f (lpar a09
   CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
   ===> cpsyntax system config f (lpar a0a
   CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
   ```

   After you confirm that no syntax errors occurred, you can release and detach the `PMAINT CF0`:

   ```
   ===> release F (detach
   DASD 0120 DETACHED
   ```

The volumes are now formatted for paging and minidisks.

6.11.6 Attaching minidisk volumes to the system for use

Detach the volumes for minidisks from MAINT and attach them to `SYSTEM`:

```
===> detach 1567-1569 156A-156F from *
1567-1569 156A-156F DETACHED
===> attach 1567-1569 156A-156F to system
DASD 1567 ATTACHED TO SYSTEM VM1567 HYPERPAV BASE
...
```

6.11.7 Shutting down and IPLing the SSI cluster again

It is recommended that you shut down and IPL again to test the changes. Complete the following steps:

1. Log on as `MAINT` on the first SSI member.
2. Before you shut down, verify that only one page volume exists by using the **QUERY ALLOC PAGE** command. A REXX EXEC is provided to run any CP command on all members in the SSI cluster. It is named **SSICMD EXEC**. Use it to issue the **QUERY ALLOC PAGE** command across the SSI cluster. The results of SSICMD EXEC are shown in Example 6-14:

   ```
   ===> ssicmd query alloc page
   ```

   Example 6-14 Results of SSICMD EXEC

   ```
   RDBKZVMG:
                   EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
   VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
   ------ ---- ---------- ---------- ------ ------ ------ ----
   VP155E 155E          1      10016  1761K   1109   1283   1%
                                     ------ ------        ----
   SUMMARY                            1761K   1109          1%
   USABLE                             1761K   1109          1%

   RDBKZVMH:
                   EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
   VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
   ------ ---- ---------- ---------- ------ ------ ------ ----
   VP1562 1562          1      10016  1761K   7182   7674   1%
                                     ------ ------        ----
   SUMMARY                            1761K   7182          1%
   USABLE                             1761K   7182          1%
   ```

3. Shut down and IPL the cluster again.

   In 6.9, “Shutting down and IPLing the SSI cluster again” on page 183, this task was accomplished manually:

   ```
   ===> shutdown
   ...
   ```

4. If you are using a 3270 emulator, you lose your session. If you watch the HMC, the SSI member LPARs immediately turn from white to green, then return to white after a minute or so.
5. After the system comes back, log on as `MAINT`.
6. Use the **SSICMD EXEC** again to issue the **QUERY ALLOC PAGE** command across the SSI cluster:

   ```
   ===> ssicmd query alloc page
   ```

You now see the new paging volumes on each of the members. The output shows two paging volumes on each SSI member that consist of 1761 KB pages each, or approximately 6.9 GB of page space (one page is 4 KB).

In total, you will have 3522 KB per member, or about 13 GB of page space. This amount is sufficient for the setup that is described in this book, but you must monitor the use of pages as an ongoing activity

6.12 Enabling z/VM basic system automation

Next, enabling basic system automation is described.

> **Note:** If you need more functions than are described in this section, see 4.3, "Operations Manager" on page 88.

6.12.1 Configuring AUTOLOG1's PROFILE EXEC

During the normal IPL process, a Virtual Service Machine (VSM) that is called `AUTOLOG1` is automatically logged on. For clarity, a normal IPL in this case is any time that the `NOAUTOLOG` parameter is not specified. The `PROFILE EXEC` for `AUTOLOG1` is run when CMS IPLs. By using this file, perform the following tasks:

1. Set OPERATOR as a secondary console for TCPIP and DIRMAINT.
2. Limit the minidisk cache with the **`SET MDC`** command.
3. Enable the memory overcommit option.

   Because `AUTOLOG1` is now a multiple configuration virtual machine (`IDENTITY`), one virtual machine is on each member. To configure the `AUTOLOG1 PROFILE EXEC`, complete the following steps:

   a. Log on to **`AUTOLOG1`**, but instead of pressing the Enter key at the `VM READ` prompt, type **`acc (noprof`** and then press Enter to log on to this ID but it will prevent the **`PROFILE EXEC`** from running:

   ```
   LOGON AUTOLOG1
   z/VM Version 6 Release 3.0, Service Level 1301 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES:   NO RDR,   NO PRT,   NO PUN
   LOGON AT 11:13:28 EDT WEDNESDAY 04/08/15
   z/VM V6.3.0      2015-01-21 10:11:10 EDT
   ===> access (noprofile
   ```

   b. Make a copy of the original `PROFILE EXEC`:

   ```
   ===> copy profile exec a = execorig =
   ```

   c. Edit the `PROFILE EXEC` and add the following lines in bold in the `Customer processing` stanza. If you do not plan to use the memory overcommit feature, omit that line:

   ```
   ===> xedit profile exec
   ====> /Customer
   ...
   /*********************************************************************/
   /* Customer processing can be added here                             */
   /*********************************************************************/
   "PIPE CP XAUTOLOG TCPIP"                /* AUTOLOGON TCPIP VSM     */
   "CP SET MDC STOR OM 256M"               /* LIMIT MDISK CACHE 256M  */
   "CP SET SRM STORBUF 300% 250% 200%"     /* VSM MEMORY OVERCOMMIT   */
   /* ------ ROUTE SCIF MESSAGES TO PROP FOR SYSLOG OR HANDLING ------ */
   "PIPE CP SET SECUSER DIRMAINT OPERATOR"  /* SEND DIRMM MSGS TO PROP */
   "PIPE CP SET SECUSER DIRMSAT1 OPERATOR"  /* SEND DIRMS MSGS TO PROP */
   "PIPE CP SET SECUSER DIRMSAT2 OPERATOR"  /* SEND DIRMS MSGS TO PROP */
   "PIPE CP SET SECUSER DIRMSAT3 OPERATOR"  /* SEND DIRMS MSGS TO PROP */
   ...
   ```

d. Save your changes and quit XEDIT:

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

e. Perform the previous set of steps on all other members in the SSI cluster.

The `PROFILE EXEC` on `AUTOLOG1 191` disk must be configured for all members in the SSI.

6.12.2 Configuring and enabling the programmable operator facility

The programmable operator facility (PROP) increases the efficiency of z/VM system operation and allows the operation of virtual guest systems in a distributed processing environment. PROP intercepts all messages and requests that are directed to the z/VM `OPERATOR` virtual machine and compares them against a *routing table,* which is a structured-format CMS file.

When a match occurs, the defined action is performed. If no match occurs, no action is performed. Certain messages are logged. Other messages are acted on automatically. Other messages are sent to an actual operator's console that is called the *logical operator console* for human intervention.

The major benefit of PROP is that the real system operator sees only important messages, while all messages are recorded for auditing.

Complete the following steps to enable PROP:

1. Log on as MAINT720 if you are not already.
2. Add a minidisk to the operator ID to use for PROP with the command:

```
===> dirmaint for operator amdisk
```

3. Complete the DirMaint AMDISK panel, as shown in Figure 6-6.

```
--------------------------------DirMaint AMDISK----------------------------
 To add a new minidisk to a user definition, fill in the following:
   Minidisk Address ===> 0692     Device Type ===> 3390
 Fill in one of the following rows:
   Explicit Start ===>           Size ===>             Volser ===>
   AUTOV                         Size ===>             Volser ===>
   VBLK   Blksize ===>         Blocks ===>             Volser ===>
   AUTOG                         Size ===> 002        Grpname ===> POOL1
   GBLK   Blksize ===>         Blocks ===>            Grpname ===>
   AUTOR                         Size ===>             Region ===>
   RBLK   Blksize ===>         Blocks ===>             Region ===>
   T-DISK                        Size ===>
   TBLK   Blksize ===>         Blocks ===>
   V-DISK                        Size ===>
   VDBS   Blksize ===>         Blocks ===>
   DEVNO           Real Device Number ===>
 Optionally fill in:
   Link Mode ===> R
   BLKSIZE   ===>         LABEL ===> OPE692
   PWS  Read ===>         Write ===>         Multi ===>
(passwords)
```

Figure 6-6 Complete the DirMaint AMDISK panel

4. After you complete the fields as shown, press **PF5** to submit. You see the following messages:

```
DVHXMT1191I Your AMDISK request has been sent for processing to DIRMAINT...
DVHSHN3430I AMDISK operation for OPERATOR address 0692 has finished
```

Alternatively, you can issue the following command instead of using the DirMaint AMDISK panel:

```
===> dirmaint for operator amdisk 0692 3390 autog 002 pool1 RR label OPE692
```

Regardless of whether you use the panel or the line command, `DVHSHN3430I` indicates that the request completed successfully.

5. Access the OPERATOR 191 disk as file mode T and the new OPERATOR 692 disk as file mode U by using VMLINK:

```
===> vmlink operator 191 < F191 T MW >
DMSVML2060I OPERATOR 191 linked MW as F191 file mode T
===> vmlink operator 692 < 0692 U MW >
DMSVML2060I OPERATOR 692 linked MW as 0692 file mode U
```

6. Copy the sample routing table (`PROP RTABLE`) file from the CMS 190 minidisk to the newly linked U disk. Because the PROP RTABLE file is mode 5 (hidden), you must access the 190 disk as C/A to copy the file. The target for the copy is **`U1`** and not U:

```
===> access 190 C/A
DMSACC724I 190 replaces C (2CC)
DMSACP723I C (190) R/O
DMSACP725I 190 also = S disk
Ready;
===> copy prop rtable C = = U1
===> release C
```

7. Modify the PROP routing table:

```
===> xedit prop rtable U
```

Complete the following steps:

a. Issue the following subcommands to XEDIT:

```
====> SET CASE UPPER
====> SET NUM ON
```

b. Locate the LGLOPR statement:

```
====> /LGLOPR
```

c. Replace “OPERATOR” with “LGLOPR” and remove the string “HOSTNODE” so that the result is similar to Example 6-15.

Example 6-15 PROP CONFIGURATION after changes

```
00001 *        ----- SPECIFY THE PROP CONFIGURATION -----
00002
00003 * IDENTIFY THE LOGICAL OPERATOR
00004
00005 LGLOPR LGLOPR
00006
```

d. Delete lines 26 - 31 by using the block-delete prefix command **`DD`** as shown. Type over 00026 with DD. Then, move the cursor to line 31 and type over 00031 with DD, as shown in Example 6-16 on page 199. Press **Enter** to delete the lines.

Example 6-16 Delete multiple lines by using block-delete

```
dd026 /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
dd031 /DROP                  21 25 3
```

e. Isolate all lines that contain the string “PROPNODE” and then delete them:

```
====> ALL/propnode
====> delete *
```

f. Isolate all lines that contain the string “NCCF” and then delete them:

```
====> ALL/NCCF
====> delete *
```

g. Clear the selection filter and save the changes that you made so far:

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

h. Move the current line to the line before “SEND ALL OTHER TRAPPED DATA TO THE LOGICAL OPERATOR” by moving to the bottom and then moving up four lines:

```
====> bottom
====> up 4
```

i. Set the case back to MIXED so that XEDIT retains uppercase and lowercase characters on the lines that you are about to enter. Then, enable INPUT mode:

```
====> set case mixed
====> input
```

j. Add these lines. PROP parses this file one line at a time and expects specific characters at specific columns. You must keep entries in their correct columns and keep characters in mixed case, as shown in Example 6-17.

Example 6-17 New routing table entries for PROP

```
*----------------------- --- --- -- -------- -------- -------- --------
* 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
*----------------------- --- --- -- -------- -------- -------- --------
```

k. Save the changes and quit XEDIT:

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

8. Make a backup of the original OPERATOR PROFILE EXEC. Then, copy the OPERATOR PROFEXEC from SFS to OPERATOR 191 as the new PROFILE EXEC:

```
===> copy profile exec T profile origexec T (olddate
===> copy operator profexec M profile exec T (olddate replace
```

9. Release and detach the OPERATOR 191 and 692 disks:

```
===> release T ( detach
===> release U ( detach
```

10. Link OPERATOR 191 to LGLOPR as 192 in the user directory entry:

```
===> dirmaint for lglopr link operator 0191 0192 RR
DVHREQ2289I Your LINK request for LGLOPR at * has completed; with RC = 0.
```

11. Set OPERATOR as the secondary user for TCPIP, EREP, MAINT, and MAINT720:

```
===> dirmaint for TCPIP COMMAND ADD 010 SET SECUSER OPERATOR
===> dirmaint for EREP COMMAND ADD 010 SET SECUSER OPERATOR
===> dirmaint for MAINT COMMAND ADD 010 SET SECUSER OPERATOR
===> dirmaint for MAINT720 COMMAND ADD 010 SET SECUSER OPERATOR
DVHREQ2289I Your COMMAND request for ... at * has completed; with RC = 0.
```

12. Log off MAINT720 but hold your connection open so that you can immediately log on as OPERATOR:

```
===> logoff hold
```

13. Log on as OPERATOR.

14. Edit PROFILE EXEC:

```
===> xedit profile exec
```

Edit the comment line at the top to indicate today's date and your ID by changing the bold italicized text, as shown in Example 6-18.

Example 6-18 PROFILE EXEC for operator and surrogate operator IDs

```
/*** OPERATOR/OP1 PROFILE EXEC A    -- MOD 2020-09-19 PWNOVAK ***/
ADDRESS COMMAND
'SYNONYM SYN'
'CP TERMINAL MODE VM'
'CP SPOOL CONSOLE TO * START NAME 'USERID()' CONSLOG'
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE BACKWARD'
'CP SET PF23 RETRIEVE FORWARD'
'CP SET PF24 RETRIEVE BACKWARD'
'ACCESS 692 D'
'IDENTIFY (LIFO'
PARSE UPPER PULL VMUSER . LOCNODE . RSCSNAME
'VMFCLEAR'
SAY '------ z/VM PROGRAMMABLE OPERATOR (PROP) ------------------'
SAY 'OPERATOR CONTROL TRANSFERRING TO PROP FACILITY AT 'LOCNODE
SAY ' '
SAY 'PROP NOW INITIALIZING AND THEN DISCONNECTING THIS TERMINAL.'
```

```
SAY '------------------------------------------------------------'
'CP SLEEP 2 SEC'
'EXEC PROPST PROP DISCONN'
```

15. Save the changes and quit XEDIT:

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

16. Remain logged on as OPERATOR but launch an additional 3270 session and log on as **LGLOPR**.
17. Position the 3270 sessions for OPERATOR and LGLOPR so that you can see both sessions at the same time.
18. In the session for OPERATOR, issue the command to IPL CMS:

    `===> **ipl cms**`

19. You see messages in both consoles. The OPERATOR console states that PROP is starting and then disconnects your session. The LGLOPR console notifies you that PROP initialized with the following message:

    `* MSG FROM OPERATOR: PROP running with routing table PROP RTABLE D1`

20. Repeat steps 11 - 15 for every other node in the SSI cluster.
21. You can leave the LGLOPR console open if you want to continue observing messages as you proceed through the next tasks. If not, issue the **LOGOFF** command for LGLOPR.

    Important: After PROP is running, do not log on as OPERATOR except when necessary. If you log on as OPERATOR, you must issue the STOP command *before* you can issue commands other than the commands that control PROP.

6.13 z/VM 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 (which is created from the source) on a CP-formatted disk.

The source form of the user directory consists of directory statements that define the CP volume on which the object directory is created and the characteristics of each virtual machine that runs on z/VM system.

Figure 6-7 shows an overview of definitions in the z/VM directory for guests with single configuration and multiple configurations:

- Single-configuration z/VM user ID

  A single-configuration VM definition consists of a user entry and any included profile entry. For example, you can specify a single-configuration virtual machine as EDIALVES and log on to a z/VM system as EDIALVES. In an SSI cluster, the VM can be logged on to only one SSI member at a time. Your Linux guests are always defined as single users and they can be relocated from one SSI-member to another using the Live Guest Relocation feature of SSI.

  Note: For more information, see *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039.

- Multi-configuration z/VM user ID

  A multi-configuration z/VM user ID definition consists of an identity entry and all associated subconfiguration entries (SUBCONFIG in BUILD ON z/VM DirMaint statement). In an SSI environment, this definition allows multiple instances, which enables the user ID to be logged on concurrently to multiple members of the SSI cluster (see Figure 6-7).

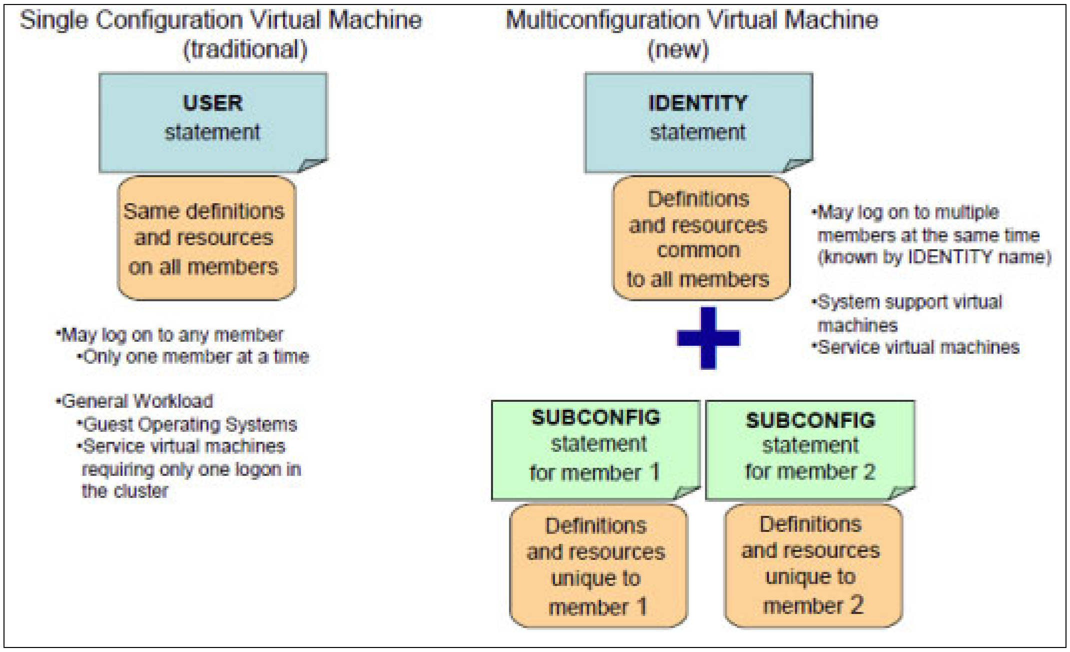


Figure 6-7 SSI User directory example

6.13.1 z/VM User Directory PROFILEs

A PROFILE entry in the z/VM User Directory is an object that defines defaults to be set and used. One PROFILE can be used by many USER, IDENTITY, or SUBCONFIG entries.

A PROFILE is easily used: add one line that is an `INCLUDE` ***PROFNAME*** statement as the second line of any USER, IDENTITY, SUBCONFIG, or PROTODIR entry that uses that PROFILE.

Consider the following points about User Directory PROFILEs:

- Each USER, IDENTITY, SUBCONFIG, or PROTODIR entry can use only one PROFILE. If you attempt to send a modified directory entry to DirMaint with two PROFILE statements in it, you will receive an error.
- In each USER, IDENTITY, SUBCONFIG, or PROTODIR entry, any statements that are listed on lines underneath the `INCLUDE` ***PROFNAME*** statement will override values from the PROFILE.

  For example, you are creating NEWUSER and including the TCPCMSU PROFILE that is supplied by IBM. You include a statement for NEWUSER to IPL CMS with auto carriage return. This statement overrides the IPL statement that is listed in the TCPCMSU PROFILE.
- No restrictions exist on the number of PROFILE entries that can be created in the z/VM user directory.

You might see that you create PROFILE entries for certain types of Linux virtual machines as time passes. For example, you might create profiles that are similar to the following examples:

– For five sizes of WebSphere Application Server in your IT Service Management (ITSM) or Information Technology Infrastructure Library (ITIL) models:

  `LNXPWASA`, `LNXPWASB`, `LNXPWASC`, and `LNXPWASD`

– For an IBM Domino Application Server that runs mail versus traditional applications:

  `LNXPDASA` and `LNXPDASB`

– For Oracle Application Server with and without Real Application Clusters (RAC):

  `LNXPORAA` and `LNXPORAB`

– For four sizes of Apache/IBM HTTP Server or Nginx in your ITSM or ITIL models:

  `LNXPWEBA`, `LNXPWEBB`, `LNXPWEBC`, and `LNXPWEBD`

Tip: Avoid directory ambiguity by adhering to a naming standard for your PROFILE entries. Use **`LNXP....`** for Linux PROFILE entries and use **`CMSP....`** for CMS. You will always easily identify what is a PROFILE versus a USER, IDENTITY, or PROTOTYPE

We create a profile for all Linux virtual machines that is called LNXPDFLT. We also explain how to review the PROFILE entries supplied by IBM which ship with z/VM, and any custom entries you might create.

Important: Do not delete or modify the PROFILE entries that are supplied by IBM. Various Virtual Service Machines and other z/VM system internals rely on the specific parameters supplied by IBM for proper operation. Use the instructions in this section to create your own custom profiles and tailor them to suit your needs.

Creating the Linux default profile

We create a PROFILE named **`LNXPDFLT`** for use with all of our Linux virtual machines. The values that are provided are reasonable defaults for most types of Linux workloads.

Complete the following steps:

1. Log on as `MAINT or MAINT720`.
2. Determine the number of active physical processors with the **`QUERY PROCESSORS`** command:

```
===> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
```

3. Create a file that is named LNXPDFLT DIRECT A and populate it by using the contents from Figure 6-8. The comments that are shown to the right of the dual asterisks (**) explain functionality and are optional:

   ===> **XEDIT LNXPDFLT DIRECT A**

```
PROFILE LNXPDFLT
***COMMON LINUX DIRMPROFILE******************************************
 COMMAND SET SECUSER OPERATOR
 COMMAND SET RUN ON
 COMMAND TERM HOLD OFF
 COMMAND TERM MORE 001 000
 COMMAND SCRE CPO WHI NON
 COMMAND SCRE STA GRE REV
 COMMAND SET VSWITCH VSW1 GRANT &USERID
 COMMAND DEFINE NIC 0600 TYPE QDIO
 COMMAND COUPLE 0600 TO SYSTEM VSW1
  CPU 00 BASE
  CPU 01
  DATEFORMAT ISODATE
  IPL CMS PARM FILEPOOL LNX AUTOCR
  MACHINE ESA 8
  IUCV ALLOW
  OPTION CHPIDV ONE
  CONSOLE 0009 3215 T
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
LINK MAINT    0190 0190 RR
LINK MAINT    019D 019D RR
LINK MAINT    019E 019E RR
LINK TCPMAINT 0592 0592 RR
```

Figure 6-8 Contents of the LNXPDFLT PROFILE that is created

The following information refers to Figure 6-8:

– The three **COMMAND** lines give the virtual machine access to virtual switch VSW1 at logon when the virtual machine is created, which precludes the need to add a VSWITCH GRANT statement each time that a Linux virtual machine is created.

– The two **CPU** lines define two virtual CPUs. It is recommended to set the number of virtual CPUs to no more than the number of available CPUs shown from **Q PROC**.

– The **MACHINE** statement sets the virtual machine type to ESA with a maximum of eight VCPUs. Even if your hardware does not have eight Integrated Facility for Linux processors (IFLs), it is alright to set the maximum value to **8** to allow growth.

– The **IUCV ALLOW** line allows virtual machines to connect to other virtual machines, such as the Linux Terminal Server, by using the inter-user communication vehicle (IUCV).

– The **OPTION CHPIDV ONE** line allows virtual machines to be relocated between SSI members.

4. Save your changes, send the entry to DirMaint for processing, and clean up the temporary files:

```
====> file
===> dirmaint add lnxpdlft
DVHREQ2289I Your ADD request for LNXPDFLT at ENDVM363 has completed; with RC=0.
===> erase * profile A
```

5. If you decide in the future that you want to change only one item in the LNXPDFLT profile, you can use individual DirMaint line commands, for example:

```
DIRMAINT FOR LNXPDFLT COMMAND ADD 001 DEFINE STORAGE 1G STANDBY 1G
DIRMAINT FOR LNXPDFLT CRYPTO APVIRT
```

Reviewing the PROFILE entries that are supplied by IBM

The following profiles are shipped with z/VM:

- **IBMDFLT**
- **TCPCMSU**
- **TCPGCSU**
- **TCPSSLU**
- **CMSDFLT**

> **Note:** Modifications made to the above entries supplied by IBM are not recommended. System components may rely on the default values, and changes can result in unexpected and unwanted results. If you want to use one of these profiles, but want to alter the contents you should make a copy of the profile with a different name and use that instead.

You can review the contents of a PROFILE by using the DirMaint REVIEW and PEEK commands:

```
===> dirmaint for ibmdflt review
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT ...
DVHREQ2288I Your REVIEW request for IBMDFLT at * has been accepted.
RDR FILE 0347 SENT FROM DIRMAINT PUN WAS 0706 RECS 0020 ...
...
===> peek 0347
```

The contents are similar to Figure 6-9.

```
0347     PEEK     A0  V 80  Trunc=80 Size=16 Line=0 Col=1 Alt=0
File IBMDFLT DIRECT from DIRMAINT at ENDVM363 Format is NETDATA.
* * * Top of File * * *
   PROFILE IBMDFLT                                              ...
   CONSOLE 0009 3215 T                                          ...
   SPOOL 000C 2540 READER *                                     ...
   SPOOL 000D 2540 PUNCH A                                      ...
   SPOOL 000E 1403 A                                            ...
   LINK MAINT 0190 0190 RR * CMS SYSTEM DISK                    ...
   LINK MAINT 019D 019D RR * HELP DISK                          ...
   LINK MAINT 019E 019E RR * PRODUCT CODE DISK                  ...
   LINK MAINT 0402 0402 RR                                      ...
   LINK MAINT 0401 0401 RR                                      ...
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20140227 CRC"·      ...
DVHREV3356I The following are your user option settings:
DVHREV3356I Links DISABLED Logging ON RcvMsg ON Smsg OFF NeedPW ON Lang AMENG
* * * End of File * * *

1= Help      2= Add line  3= Quit      4= Tab      5= Clocate     6= ?/Change
7= Backward  8= Forward   9= Receive  10= Rgtleft 11= Spltjoin   12= Cursor

====> discard
                                                     X E D I T  1 File
```

Figure 6-9   Contents of IBMDFLT PROFILE entry

When your review is complete, discard the file so that it does not occupy spool space unnecessarily. Peek will close and the file will be discarded:

```
====> discard
File IBMDFLT DIRECT has been discarded.
```

6.13.2 Role-based access controls and CP privilege classes

In z/VM, a user can have no privileges or be assigned to one or more privilege classes. Each privilege class represents a subset of CP commands that the system permits the user to enter.

Each privilege class, sometimes called *CP privilege class*, is defined around a particular job or set of tasks, which creates an area outside of which the user cannot go. It is common for a user to be assigned to more than one CP privilege class. Users cannot enter commands in privilege classes to which they are not assigned.

Tip: It is also possible to create an override of privilege classes to meet the enterprise security policy according to the roles of your installation.

The following CP privilege classes are available:

- Privilege class A

  This class is the primary system operator. The system operator is among the most powerful and privileged of all z/VM users. The system operator is responsible for the system's availability and its resources. The system operator also controls accounting, broadcasts messages, and sets performance parameters.

- Privilege class B

  This class is the system resource operator. The system resource operator controls the allocation and de-allocation of real resources, such as memory, printers, and DASD. The system resource operator does not control any resource that is controlled by the system operator or the spooling operator.

- Privilege class C

  This class the system programmer. The system programmer updates the functions of the z/VM system and can change real storage in the real machine.

- Privilege class D

  This class is the spooling operator. The spooling operator controls spool files and real unit record devices, such as punches, readers, and printers.

- Privilege class E

  This class is the system analyst. The system analyst has access to real storage and examines dumps to make sure that the system is performing as efficiently and correctly as possible.

- Privilege class F

  This class is the IBM service representative who diagnoses and solves problems by examining and accessing real input and output devices and the data they handle.

- Privilege class G

  This class is the CMS general user. This class is the most prevalent and innocuous of the CP privilege classes. The commands that privilege class G users can enter affect only their own VM user IDs.

Tip: Linux servers are generally created with class G or less.

For more information about enhancing security by further restricting the z/VM privileges that are granted to each Linux guest, see *Running Linux Guest in less than CP Privilege Class G*, REDP-3870.

6.13.3 Creating and using z/VM User Directory prototypes

You create two prototype directory entries (PROTODIR): One prototype directory entry is for CMS, and one prototype directory entry is for Linux so that you can quickly and easily add new SCVMs to the system. Prototypes are essentially a template, which is used to build a new virtual service machine ID with a standard set of resources.

IBM supplies two sample prototypes: one for a for typical CMS virtual machine, and one for a sample Linux virtual machine. Complete the following steps:

1. To verify that these are present, issue the following commands as **MAINT720** on any cluster member. You need to issue these commands only one time during the initial setup of your z/VM cluster:

   ```
   ===> DIRMAINT FOR DIRMAINT CMS LISTFILE * PROTODIR *
   DVHREQ2288I Your CMS request for DIRMAINT at * has been accepted.
   DVHCMS3868I CMS PROTODIR E2
   DVHCMS3868I LINUX PROTODIR E2
   DVHCMS3868I CMS PROTODIR G2
   DVHCMS3868I LINUX PROTODIR G2
   ```

   If you see the response above, skip Step 2.

2. If you do *not* see the response that are shown in Step 1, issue the following commands. After they complete, re-issue the command from Step 1 to verify they are now present:

   ```
   ===> DIRMAINT CMS COPYFILE CMS DATADVH D = PROTODIR E (OLDDATE
   ===> DIRMAINT CMS COPYFILE LINUX DATADVH D = PROTODIR E (OLDDATE
   ```

We use both of these default files.

6.13.4 Creating CMSPROTO

DirMaint ships with a basic CMS PROTODIR that is a good starting point. Use this process to customize this file in preparation for use.

While you are logged in as **MAINT** or **MAINT720** on any node in the cluster, complete the following steps:

1. Start a **DIRMAINT SEND** request for the **CMS PROTODIR** on file, and then receive the file to your A disk with the new filename of **CMSPROTO** for editing:

   ```
   ===> dirmaint send cms protodir
   DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT ...
   ...
   RDR FILE 0015 SENT FROM DIRMAINT PUN WAS 3485 RECS ...
   ...
   ===> receive 0015 cmsproto = A
   FILE CMSPROTO PROTODIR A2 created from CMS PROTODIR E2 received from DIRMAINT
   ...
   ```

2. Edit the file so that it looks like Example 6-19:

   ```
   ===> xedit cmsproto protodir A
   ```

   Example 6-19 Edit the file

   ```
   USER CMSPROTO NOLOG 32M 64M G
      INCLUDE TCPCMSU
      IPL CMS PARM FILEPOOL VMSYSU AUTOCR
      COMMAND SET RUN ON
   ```

```
DATEFORMAT ISODATE
AMDISK 0191 3390 AUTOG 00004 USRWORK MR
```

3. Send the new PROTODIR file to DirMaint for filing:

```
===> dirmaint file cmsproto protodir A
...
DVHRCV3821I File CMSPROTO PROTODIR A has been received; RC = 0.
DVHXMT1191I Your FILE request ... has completed; with RC = 0.
```

4. Erase the temporary copy of the PROTODIR file from the local A disk:

```
===> erase cmsproto protodir A
```

Your new prototype directory template is now ready for use. In the future, if you want to modify this new prototype, follow the steps in the next section.

Modifying a z/VM User Directory prototype

Complete the following steps to modify a prototype that you created:

1. Request the prototype record. In this case, we use **CMSPROTO** as an example:

```
===> DIRMAINT SEND CMSPROTO PROTODIR
```

2. Substitute ***1234*** with the I reader file number each time:

```
RECEIVE 1234 = = A (OLDD REP
```

To continue, run the following commands:

– XEDIT CMSPROTO PROTODIR A
– DIRMAINT FILE CMSPROTO PROTODIR A
– ERASE CMSPROTO PROTODIR A

> **Tip:** It is important that you erase the temporary copies of prototype directory files when you are finished with them. Although it is tempting to leave them on the A disk, if multiple people work with them and log on as MAINT from different nodes in the cluster, it is easy to assume that the local copy is current and overwrite previous changes. By always asking DirMaint for the latest copy on file and coordinating your efforts with other z/VM system programmers, you reduce the likelihood that this problem will happen.

Creating a user ID by using CMSPROTO

You now create a user ID for yourself that you will use to log on. You will also use this user ID to configure LOGONBY.

While you are logged in as **MAINT** on any node in the cluster, issue the following dirmaint command. In this example, the new directory entry that is added is ***pwnovak*** with a temporary password of ***need2chg***:

```
===> dirmaint add pwnovak like cmsproto pw need2chg
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT ...
...
DVHREQ2289E Your ADD request for PWNOVAK at * has completed; with RC = 0.
```

You can now log in as this new user.

Creating SUBPRO-1

While you are logged in as **MAINT** or **MAINT720** on any node in the cluster, complete the following steps:

1. Create a file that is named **SUBPRO-1 PROTODIR A**. It will contain only one line:

```
===> xedit subpro-1 protodir A
====> input SUBCONFIG SUBPRO-1
```

2. Modify and save new copies for each member in your cluster. If your SSI cluster has:

   – One member:

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

   – Two members:

```
====> save SUBPRO-1 PROTODIR A
====> c/1/2/* * # save SUBPRO-2 PROTODIR A
====> file
```

   – Three members:

```
====> save SUBPRO-1 PROTODIR A
====> c/1/2/* * # save SUBPRO-2 PROTODIR A
====> c/2/3/* * # save SUBPRO-3 PROTODIR A
====> file
```

   – Four members:

```
====> save SUBPRO-1 PROTODIR A
====> c/1/2/* * # save SUBPRO-2 PROTODIR A
====> c/2/3/* * # save SUBPRO-3 PROTODIR A
====> c/3/4/* * # save SUBPRO-4 PROTODIR A
====> file
```

3. Send the new PROTODIRs to DirMaint for filing:

```
===> dirmaint file SUBPRO-1 protodir A
...
DVHRCV3821I File SUBPRO-1 PROTODIR A has been received; RC = 0.
DVHXMT1191I Your FILE request ... has completed; with RC = 0.
===> dirmaint file SUBPRO-2 protodir A
...
DVHXMT1191I Your FILE request ... has completed; with RC = 0.
===> dirmaint file SUBPRO-3 protodir A
===> dirmaint file SUBPRO-4 protodir A
```

4. Erase the temporary working copies of the protodir from the local A disk:

```
===> erase subpro* protodir A
```

Your new subdirectory prototype directory templates are now ready for use.

6.13.5 Creating LNXPROTO

DirMaint ships with a basic `Linux PROTODIR` that is a good starting point. Use the following steps to customize this file in preparation for use.

While you are logged in as **MAINT** on any node in the cluster, complete the following steps:

1. Start a **DIRMAINT SEND** request for the default file, then receive the file to your A disk as a new file for editing by specifying the name:

   ```
   ===> dirmaint send linux protodir
   DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT ...
   ...
   RDR FILE 0018 SENT FROM DIRMAINT PUN WAS 3485 RECS ...
   ...
   ===> receive 0018 lnxproto = A
   ```

2. Edit the file by using the following command so that it looks like Example 6-20:

   ```
   ===> xedit lnxproto protodir A
   ```

   Example 6-20 Contents of LNXPROTO with 5008 cylinder minidisk

   ```
   USER LNXPROTO NOLOG
    INCLUDE LNXPDFLT
     AMDISK 0100 3390 AUTOG 5008 POOL1 MR
   ```

   In this example, the value of **5008** indicates that the new Linux virtual machines that are created by using this PROTODIR are given a minidisk of 5008 cylinders from POOL1 that we defined earlier in the DirMaint EXTENT CONTROL file. The value of 5008 cylinders is one half of a 3390 model 9 DASD.

   If you want to give each of your virtual machines a full-pack minidisk by default, you must change the value to 10016, as shown in Example 6-21. If you do not want to include a default 0100 minidisk at all, omit this line and use **DIRMAINT AMDISK** later to generate a 0100 minidisk for each virtual machine that you create.

   Example 6-21 Contents of LNXPROTO with 10016 cylinder minidisk

   ```
   USER LNXPROTO NOLOG
    INCLUDE LNXPDFLT
     AMDISK 0100 3390 AUTOG 10016 POOL1 MR
   ```

   DIRMAINT AMDISK is described in section 6.17, “Creating identity LNXADMIN for Linux administration” on page 246 and with greater detail in section 11.4.10, “Adding a minidisk to a user or identity” on page 367.

3. Send the new PROTODIR to DirMaint for filing:

   ```
   ===> dirmaint file lnxproto protodir A
   ...
   DVHRCV3821I File LNXPROTO PROTODIR A has been received; RC = 0.
   DVHXMT1191I Your FILE request ... has completed; with RC = 0.
   ```

4. Clear the temporary copy of the protodir from the A disk:

   ```
   ===> erase lnxproto protodir A
   ```

Your new prototype directory template is now ready for use.

Listing of all z/VM User Directory prototypes

To obtain a full list of prototypes which are known to DirMaint, issue the following command:

```
===> dirmaint for dirmaint cms listfile * protodir *
DVHXMT1191I Your CMS request has been sent for processing to DIRMAINT ...
...
 DVHCMS3868I CMSPROTO PROTODIR A2
 DVHCMS3868I LNXPROTO PROTODIR A2
 DVHCMS3868I SUBPRO-4 PROTODIR A1
 DVHCMS3868I SUBPRO-1 PROTODIR A1
 DVHCMS3868I SUBPRO-2 PROTODIR A1
 DVHCMS3868I SUBPRO-3 PROTODIR A1
 DVHCMS3868I CMS      PROTODIR E2
 DVHCMS3868I LINUX    PROTODIR E2
 DVHCMS3868I CMS      PROTODIR G2
 DVHCMS3868I LINUX    PROTODIR G2
 DVHREQ2289I Your CMS request for DIRMAINT at * has completed; with RC = 0.
...

or use PROTOLST EXEC that was downloaded, as described in 6.11.2, "Copying the
utilities to Shared File System file pools" on page 186,
```

Note: You see more than one line for the CMS and LINUX protodirs. This is expected and is not an error.

Reviewing contents of a z/VM User Directory prototype

You must know the name of the prototype to review its contents. We obtained a full listing of all prototypes as described in, "Listing of all z/VM User Directory prototypes".

Using LNXPROTO as an example, the following command shows you the contents of the LNXPROTO prototype:

```
===> dirmaint for dirmaint cms type lnxproto protodir a
DVHXMT1191I Your CMS request has been sent for processing to DIRMAINT ...
...
DVHREQ2288I Your CMS request for DIRMAINT at * has been accepted.
 DVHCMS3868I USER LNXPROTO NOLOG
 DVHCMS3868I     INCLUDE LNXPDFLT
 DVHCMS3868I     AMDISK 0100 3390 AUTOG 5008 POOL1 MR
 DVHREQ2289I Your CMS request for DIRMAINT at * has completed; with RC = 0.
```

6.13.6 Creating a time-based virtual service machine named CRONSVM

Create a virtual machine that is used to run time-based activities, which are called *WAKEUPs*. This function is analogous to the root crontab in Linux. The user ID for this new virtual machine is **CRONSVM**.

Complete the following steps:

1. Log on as **MAINT** or **MAINT720** on any cluster member, if necessary.
2. Run the following commands. The password is set to **LBYONLY** and needs to stay that way. LBYONLY is analogous to setting the default shell for a Linux task ID to /bin/false or /sbin/nologin and requiring users to issue **sudo su - cronsvm** to obtain access:

```
===> dirmaint add cronsvm like cmsproto pw LBYONLY
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT ...
```

```
...
DVHREQ2289E Your ADD request for CRONSVM at * has completed; with RC = 0.
```

3. This virtual machine has special requirements, which you set by editing the directory entry:

```
===> dirmaint for cronsvm get lock
DVHREQ2288I Your GET request for CRONSVM at * has been accepted.
DVHGET3304I Directory entry  CRONSVM is now locked.
RDR FILE 1301 SENT FROM DIRMAINT PUN WAS 5037 RECS 0010 ...
...
==> receive 1301
File CRONSVM DIRECT A0 created from CRONSVM DIRECT A0 ...
```

4. Change the directory entry so that it looks like Example 6-22:

```
===> xedit cronsvm direct a
```

Example 6-22 Change the directory entry

```
USER CRONSVM LBYONLY 32M 32M ABCDEFG
   INCLUDE TCPCMSU
   ACCOUNT 3 OPERATOR
   LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNOVAK SPIEDIE
   IPL 190
   MACH XA
   LINK OPERATOR 0191 0192 RR
   LINK MAINT 0193 0193 RR
   LINK TCPMAINT 592 592 RR
MDISK 0191 .... //DO NOT ALTER THIS LINE IN YOUR FILE
*DVHOPT ....... //DO NOT ALTER THIS LINE IN YOUR FILE
```

Important: Consider the following points:

- While you are editing a directory entry that you received by using the **DIRMAINT FOR ... GET** command or the short command **DIRM FOR ... GET**, the last line of the file contains internal data that is used by DirMaint during processing.

  Do not change, delete, or move the line beginning with `*DVHOPT`.
- If you accidentally delete or modify the `*DVHOPT` line, use the XEDIT subcommand **QQUIT** to quit without saving your changes and then, restart your XEDIT session for the file. This approach is effective if you did *not* use the **SAVE** subcommand during your XEDIT session.

  If you performed an intermediate SAVE, use **QQUIT** to exit without saving any further changes, **ERASE** the locally saved directory entry from your A disk, unlock the record by issuing the command **DIRMAINT FOR ... UNLOCK**, and then, start over again.

```
===> dirmaint for cronsvm replace
PUN FILE ... SENT TO   DIRMAINT RDR AS  ...
DVHXMT1191I Your REPLACE request has been sent for processing
DVHREQ2288I Your REPLACE request for CRONSVM at * has been accepted.
...
DVHBIU3428I Changes made to directory entry CRONSVM have been placed online.
DVHREP3603I Directory entry CRONSVM is now unlocked.
DVHREQ2289I Your REPLACE request for CRONSVM at * has completed; with RC = 0.
```

5. Link to the CRONSVM 191 minidisk read/write as file mode X:

```
===> vmlink cronsvm 191 < C191 X MR >
DMSVML2060I CRONSVM 191 linked MR as C191 file mode X
```

6. Copy the VMCRON EXEC from SFS to X and review the contents. Edit the comment line at the top to indicate today's date and your ID by changing the bold italicized text that is shown in Example 6-23:

```
===> copy VMCRON EXEC M = = X (olddate
===> xedit vmcron exec X
```

Example 6-23 Sample TIMED EXEC

```
/** VMCRON EXEC (TIMED) : CRONSVM 191 - MOD 2015-04-12 YOURID **/
/* This is a sample application of the WAKEUP 'FILE' option.   */
/* This EXEC uses the WAKEUP TIMES file.                       */
/**************************************************************/
Address COMMAND
Do forever
  'MAKEBUF'
  'WAKEUP (FILE(WAKEUP)'
  if rc=3 then Do
    pull var1
    'DROPBUF'
    /* parse field 4 from the stacked wakeup times file line   */
    parse upper value var1 with asterisk reqno field1 field2 ,
                                field3 command
    if command='MSG01' then Do
      'CP MSG OPERATOR THE TIME IS NOW:' time() 'ON' date()
      'CP SLEEP 3 MIN'             /* sleep through midnight */
      END
    else
      if command><'' then Do
        if subword(command,1,1)='CMS' then
        command=subword(command,2) /* strip off cms part     */
        address CMS command        /* execute command        */
      end                          /* end of if command><''  */
  end
  else Do
    'DROPBUF'
    leave
  end                                /* end of else Do         */
end                                  /* end of Do forever loop */
exit
```

7. Create the PROFILE EXEC for CRONSVM by using the information that is shown in Figure 6-10.

```
/*** CRONSVM PROFILE EXEC A : CRONSVM 191 -- MOD 2020-09-19 PWNOVAK ***/
'CP SET RUN ON'
'CP SPOOL CONSOLE CLOSE'
'CP MSG OPERATOR LOGON 'USERID()' FOR VMCRON TIMED'
'CP SPOOL CONSOLE TO VMLOGS START NAME' USERID() ]CONSOLE]
'ACCESS 193 U'
'ACCESS 592 X'
'EXEC VMCRON'
```

Figure 6-10 CRONSVM PROFILE EXEC contents

Maintaining the spool automatically with SFPURGER

The SFPURGER utility manages spool space and spool files. SFPURGER will be set up to run unattended on the VMLOGS virtual machine. SFPURGER performs spool file maintenance by using instructions that you provide ahead of time, at intervals that you determine. You provide the instructions to SFPURGER by using options and control files, and SFPURGER records its processing in a set of output files. For complete details about the SFPURGER utility, see *z/VM CP Commands and Utilities Reference*, SC24-6175.

SVMREST handling of EREP records

The *Environmental Record Editing and Printing Program (EREP): Reference*, GC35-0152, and the *Environmental Record Editing and Printing Program (EREP): User's Guide*, GC35-0151, explain the EREP, its options, and the format of each type of EREP record. We do not cover all of the information here. To summarize, the EREP starts automatically at system IPL and tracks the activities on the system by generating record files. You use the EREP program to format and print EREP records.

When too many EREP records are queued for processing, the following message appears on the OPERATOR (or LGLOPR) console:

```
HCPCRC8083I EREP RECORD THRESHOLD HAS BEEN EXCEEDED FOR USERID EREP. CURRENTLY
00048816 RECORDS ARE ENQUEUED.
```

The VMLOGS VSM invokes a routine that is called SVMREST to attempt to prevent this situation.

6.13.7 Creating a console logs repository

On z/VM, console logs are created each time a user ID is logged on to the system. These logs are an excellent starting point when performing problem determination on an issue or to understand a specific behavior.

In this section, we describe how to create a virtual service machine that will collect and store log files that are sent to its virtual spool. It will also invoke SFPURGER and SVMREST. The installation will use the package from the z/VM downloads page, which is included in the compressed file that you downloaded earlier into SFS. This package populates a virtual machine that is named VMLOGS to act as a repository for consoles and any data file that needs to be accessible for a predetermined number of days.

Logs are spooled to the VMLOGS VSM and automatically received onto the VMLOGS 191 minidisk. Files are kept until a specified maximum age, then automatically purged.

Directing all console logs to VMLOGS creates a centralized way of monitoring system-wide activities, including Linux virtual machines.

Complete the following steps:

1. Log on as **MAINT** or **MAINT720** on any cluster member if you are not logged on.
2. Create the following directory entry by using the following command.

   ```
   ===> xedit vmlogs direct a
   ```

   Enter the information shown in Example 6-24. Enter your own user IDs on the line with LOGONBY.

   Example 6-24 VMLOGS directory entry

   ```
   USER VMLOGS LBYONLY 64M 128M ABDEG
      INCLUDE IBMDFLT
      IPL CMS PARM AUTOCR
      LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNOVAK SPIEDIE
      MACH ESA
      LINK OPERATOR 0191 0291 RR
      LINK MAINT 0193 0293 RR
      LINK TCPMAINT 592 592 RR
   MDISK 0191 3390 AUTOG 1000 USRWORK MR READ WRITE MULTIPLE
   MDISK 0193 3390 AUTOG 0005 USRWORK MR READ WRITE MULTIPLE
   ```

3. Send the new entry to DirMaint for processing:

   ```
   ===> dirmaint add vmlogs
   ...
   DVHREQ2289I Your ADD request for VMLOGS at * has completed; with RC = 0.
   ```

4. Erase the temporary directory file:

   ```
   ===> erase vmlogs direct a
   ```

5. Access the VMARC SFS directory as P and also change M to be forcerw:

   ```
   ===> access VMPSFS:MAINT720.UTILS.VMARC P (forcerw
   ===> access VMPSFS:MAINT720.UTILS M (forcerw
   ```

6. Access the new VMLOGS 193 minidisk as W:

   ```
   ===> vmlink VMLOGS 193 < F193 W MR >
   ```

7. Re-block and unpack the VMLOGS VMARC to W:

   ```
   ===> PIPE < VMLOGS VMARC P | FBLOCK 80 00 | > VMLOGS VMARC P F 80
   ===> VMARC UNPK VMLOGS VMARC P = = W
   CLEANUP  EXEC     W5. Bytes in=      880, bytes out=     1258 (   142%).
   LOGPGM   EXEC     W1. Bytes in=     5120, bytes out=    10760 (   210%).
   PROFILE  VMLOGS   W1. Bytes in=      640, bytes out=      819 (   127%).
   SFPURGER CONTROL  W2. Bytes in=     1280, bytes out=     3920 (   306%).
   SFPURGER OPTIONS  W1. Bytes in=      560, bytes out=     1440 (   257%).
   SLINK    EXEC     W1. Bytes in=      640, bytes out=     1062 (   165%).
   SVMREST  EXEC     W1. Bytes in=      800, bytes out=     1257 (   157%).
   VMLOGS   CONTENTS W1. Bytes in=     3040, bytes out=     8960 (   294%).
   VMLOGS   DIRECT   W1. Bytes in=      400, bytes out=      480 (   120%).
   VMLOGS   PARMS    W5. Bytes in=      560, bytes out=      960 (   171%).
   ```

8. Erase the unnecessary directory entry from W, then release and detach W:

   ```
   ===> erase vmlogs direct W
   ===> release W (detach
   ```

9. Issue the following command to start VMLOGS with the **XAUTOLOG** command and the **SYNC** option that returns control to MAINT/MAINT720, and sets MAINT/MAINT720 to be the secondary user. This way, VMLOGS does not have to be logged on to, but you can see its console output:

```
===> xautolog vmlogs sync # set secuser vmlogs *
AUTO LOGON  ***       VMLOGS USERS = 15
Ready;
HCPCFX6768I SECUSER of VMLOGS initiated.
Ready;
```

   Watch for errors and check to ensure that everything appears to start successfully, then remove the secondary user messages:

```
===> set secuser vmlogs off
VMLOGS: HCPCFX6769I Your SECUSER terminated by MAINT720.
HCPCFX6769I SECUSER of VMLOGS terminated.
```

10. Enable the automatic logon of the ID during system start-up:

```
===> dirmaint for autolog1 xautolog add vmlogs
...
DVHREQ2289I Your XAUTOLOG request for AUTOLOG1 at * has completed; with RC=0
```

6.14 z/VM security and hardening

The following security and system hardening topics are discussed in this section:

- Use an external security manager for correct resource security
- Using LOGONBY for correct accountability
- High-level z/VM security
- Encrypting communications with SSL/TLS on z/VM

6.14.1 Using an external security manager for correct resource security

Consider the implementation of a z/VM external security manager (ESM), such as IBM Resource Access Control Facility for z/VM (RACF/VM) or CA VM:Secure. With them, you can correctly implement security policies, such as password encryption, password aging, and audit logging. If your ESM provides a Lightweight Directory Access Protocol (LDAP) interface, this interface might help to simplify the management of all your Linux virtual machines. For example, you might configure Linux to rely on LDAP through Protocol Analysis Module (PAM) to eliminate the need for individual user IDs that are created on each Linux virtual machine.

If your system processes data in a regulated industry, the use of an ESM is likely mandatory. This book covers the basic setup of RACF/VM in , “This output shows that SMAPI is running, LNXADMIN is correctly authorized to call SMAPI, and the Linux interface **smaclient** is working.” on page 340.

6.14.2 Using LOGONBY for correct accountability

Similar to how you normally configure a Linux system so that direct login by using the root or another highly privileged system account is impossible, we describe the necessary steps to provide the same function for z/VM. It is similar to configuring a Linux system so that users are required to log in as their own ID and use `sudo` to issue privileged commands.

In 6.13.2, "Role-based access controls and CP privilege classes" on page 206, we created a new ID that is named pwnovak. In our example environment, we also created individual IDs for all of the authors of this book. We now grant the new IDs the LOGONBY privilege for `LGLOPR`, `MAINT`, and `MAINT720`.

While logged on as MAINT or MAINT720, complete the following steps:

1. Issue the DirMaint `logonby` command to open the LOGONBY panel as shown in Figure 6-11 on page 218:

   ```
   ===> dirmaint for MAINT720 logonby
   ```

```
--------------------------------DirMaint LOGONBY----------------------------

 Query or update the list of users on the current LOGONBY directory
statement.

 Select one of the following:
       _ ? (Query)
       X ADD
       _ DELETE

 For ADD or DELETE, fill in one or more Userids:
    ===> EDIALVES
    ===> MAURO
    ===> LYDIAP
    ===> PWNOVAK
    ===> VIC
    ===> ________
    ===> ________
    ===> ________
```

Figure 6-11 DirMaint LOGONBY panel

   Press **F5** to proceed. You see the following messages:

   ```
   DVHXMT1191I Your LOGONBY request has been sent for processing to DIRMAINT ...
   Ready;
    DVHREQ2288I Your LOGONBY request for MAINT720 at * has been accepted.
    DVHREQ2289I Your LOGONBY request for MAINT720 at * has completed; with RC = 0.
   ```

2. Repeat the previous step for OPERATOR, LGLOPR, and MAINT:

   ```
   ===> dirmaint for operator logonby
   ...
   ===> dirmaint for lglopr logonby add edialves mauro vic pwnovak lydiap
   ...
   ```

You might also choose to bypass the use of the LOGONBY panel by using the ADD subcommand and the list of IDs to add:

```
===> dirmaint for maint logonby add edialves mauro vic pwnovak lydiap
DVHXMT1191I Your LOGONBY request has been sent for processing to DIRMAINT ...
Ready;
 DVHREQ2288I Your LOGONBY request for MAINT at * has been accepted.
 DVHBIU3450I The source for directory entry MAINT has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHRLA3891I Your DMVCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry MAINT have been placed online.
 DVHREQ2289I Your LOGONBY request for MAINT at * has completed; with RC = 0.
```

3. You might want to query the list of authorized IDs by using the following command:

```
===> dirmaint for maint logonby ?
DVHXMT1191I Your LOGONBY request has been sent for processing to DIRMAINT ...
Ready;
 DVHREQ2288I Your LOGONBY request for MAINT at * has been accepted.
 DVHLBY3331I The current LOGONBY statement in MAINT is as follows:
 DVHLBY3331I LOGONBY EDIALVES MAURO VIC PWNOVAK LYDIAP
DVHREQ2289I Your LOGONBY request for MAINT at * has completed; with RC = 0.
```

4. Log on as your new ID and change your password to a unique, secure password of your own choosing. Each person must create a password for their own ID:

```
===> dirmaint for edialves pw
DVHPWC1362R PW command is running.
DVHPWC1362R Enter your new password.  It will not be shown.  To exit
DVHPWC1362R without changing your password, just press ENTER.

// Type your new password and press Enter. Nothing shows up on the window.

DVHPWC1364R Enter your new password again, for typographical
DVHPWC1364R verification. It will not be shown.  To exit without
DVHPWC1364R changing your password, just press ENTER.

// Retype your new password and press Enter. Nothing shows up on the window.

DVHXMT1191I Your PW request has been sent for processing to DIRMAINT ...
Ready;
 DVHREQ2288I Your PW request for EDIALVEES at * has been accepted.
 DVHREQ2289I Your PW request for EDIALVES at * has completed; with RC = 0.
```

5. Log on as one of the privileged z/VM IDs by using your ID and by entering the command that is shown in Figure 6-12 on page 220. Then, press the Enter key. Completing the USERID and PASSWORD fields will not work, you must go immediately to the COMMAND field.

```
z/VM ONLINE    Welcome to the IBM z/VM Enterprise Virtualization Platform


                IBM REDBOOKS SG24-8147-00
                The Virtualization Cookbook for Linux on IBM z Systems

                                   / VV\         VVV\MM\      MM\
                                  /  VV\       VVV\ MMM\    END\
                      ZZZZZZ     /   VV\     VVV\  MMMM\  MMMM\
                         ZZ     /    VV\   VVV\   MM\MM\MM\MM\
                        ZZ     /     VV\ VVV\    MM\ MMM\ MM\
                       ZZ     /      VVVVV\     MM\  M\  MM\
                      ZZ     /       VVV\      MM\      MM\
                     ZZZZZZ /        V\       MM\      MM\

                ibm.com/vm      Built on IBM Virtualization Technology


 Fill in your USERID and PASSWORD and press ENTER
 (Your password will not appear when you type it)
 USERID   ===>
 PASSWORD ===>

 COMMAND  ===>  logon MAINT720 by edialves

                                                         RUNNING  RDBKZVMG
```

Figure 6-12 Log on by using LOGONBY

6. The panel clears and in the next panel, you see an acknowledgment that CP is going to process this log on for `MAINT720 BY PWNOVAK`. In Figure 6-13, the password for EDIALVES is now entered at the ENTER PASSWORD prompt.

```
LOGON MAINT720 BY EDIALVES
ENTER PASSWORD  (IT WILL NOT APPEAR WHEN TYPED):




CP READ   RDBKZVMG
```

Figure 6-13 LOGONBY password prompt

7. After you complete and test the passwords successfully, the passwords for OPERATOR, LGLOPR, MAINT, and MAINT720 must be changed. Change the passwords to a unique password for each ID. Only two people, such as the z/VM chief systems programmer and their immediate manager, must know these passwords. Direct log on as any of these IDs must occur in emergency situations only where LOGONBY is not possible:

```
===> dirmaint for maint pw
DVHPWC1362R PW command is running.
DVHPWC1362R Enter your new password.  It will not be shown.  To exit
DVHPWC1362R without changing your password, just press ENTER.
...
```

Another option is to change the passwords for all of these IDs to **LBYONLY**, which might be useful in situations with password change interval requirements.

> **Note:** The command syntax differs slightly to set a password to LBYONLY, AUTOONLY, or NOPASS. The following syntax is used to set one of these special reserved passwords:
>
> ```
> ===> dirmaint for ... setpw lbyonly
> ```

6.14.3 High-level z/VM security

The *z/VM Security and Integrity* paper describes the isolation and integrity of virtual machines under z/VM. It is available here.

For more information about the latest news, pertinent presentations, papers, Redbooks documents, publications, links to press articles, and pointers to online discussions, see the z/VM security page.

On this page, click **Notify Me** to be automatically notified by email when new information is available.

6.14.4 Encrypting communications by using SSL/TLS on z/VM

Correctly implementing and managing security controls for the z/VM hypervisor is a mandatory cornerstone, no matter how large or small your enterprise is. Your security posture is only as strong as the weakest point, which means that the correct encryption of traffic must be implemented at all layers. Connectivity to the hypervisor layer and well-secured guests on an unsecured hypervisor are critical exposures. Furthermore, in nearly all circumstances, encrypting traffic as a default practice is common sense. Encryption requirements might also be mandated by company policy, customers, partners, vendors, industry regulations, or governing bodies.

The Secure Sockets Layer (SSL) server provides the processing capability that allows encrypted communication between two TCP/IP connection participants, one of which is a server or client application on the local z/VM host. Dynamic SSL/Transport Layer Security (TLS) connections are supported by the following z/VM TCP/IP application servers and clients, which are updated to accommodate this support:

- TCP/IP server
- SSL server
- FTP server
- FTP client
- Telnet server (Internal to the TCP/IP server)
- Telnet client
- Simple Mail Transfer Protocol (SMTP) server

Server certificates and certificate authority (CA) certificates are stored in a certificate (key) database, which is in the z/VM Byte File System (BFS) and managed independently of the SSL server. Management is performed by using the `GSKKYMAN` utility program, which is built around the IBM Global Security Kit (GSKit). The SSL server also provides an `SSLADMIN` command interface for dynamic server operation that allows certificate database administration and server administration tasks. The setup in this book covers the use of TLS while explicitly disabling SSL v3 and lower to mitigate the heartbleed vulnerability, which is a serious vulnerability in the popular OpenSSL cryptographic software library.

Configuration of TCP/IP transport encryption consists of the following steps:

1. Update the `TCPIP PROFILE` and `DTCPARMS` files.
2. Use the `SSLPOOL` utility to create the necessary configuration changes.
3. Set up the certificate database.
4. Generate a self-signed certificate for the SSI cluster.
5. Implement Customization for Protected Communications.
6. Configure TLS Services (Dynamic SSL/TLS Connections).

Complete the following steps to update the `PROFILE` and `DTCPARMS` files:

1. Log on as **TCPMAINT**.
2. Make a backup copy of the TCP/IP configuration file, `PROFILE TCPIP D`:

   ```
   ===> copy profile tcpip d = tcpiwork = (olddate
   ```

3. Edit the TCP/IP configuration file:

   ```
   ===> xedit profile tcpip d
   ```

4. By using the contents of Example 6-26 for reference, make the following changes:

   a. Add an **SSLSERVERID** statement as a new line underneath the `LARGEENVELOPEPOOLSIZE` line.

   b. On the following line, include an **SSLLIMITS** statement to specify the total number of secure connections that are allowed and the connection limit for each SSL server.

   c. Above the `PORT` stanza, create an **INFORM** stanza to contain the user IDs to notify if any serious TCP/IP or associated issues are detected.

   d. In the `PORT` stanza, add a semicolon to the beginning of the line for `INTCLIENT` that uses port 23.

   e. Also in the `PORT` stanza, add a line underneath the last line for LDAPSRV. Replace the value ***ITSOSSIA*** with the value from the planning worksheet that you chose for your SSI cluster name:

   ```
   992  TCP INTCLIEN  SECURE ITSOSSIA ; TN3270 IntClient Server (Secure)
   ```

   f. Just below the `PORT` stanza, create another stanza for **INTERNALCLIENTPARMS** as shown. The INACTIVE statement is optional. The secondary PORT statement is also optional.

   These changes cause transport security services to start when TCP/IP is started.

   The important lines in this file are shown before the file is edited in Example 6-25, and after the file is edited in Example 6-26.

Example 6-25 TCPIP profile before SSL server modifications

```
; ------------------------------------------------------------
LARGEENVELOPEPOOLSIZE 50 16384
; ------------------------------------------------------------
   ...
; ------------------------------------------------------------
PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
   ...
  2049 UDP VMNFS               ; NFS Server
  2049 TCP VMNFS     NOAUTOLOG ; NFS Server
```

Example 6-26 TCPIP profile with SSL Server modifications shown

```
; -----------------------------------------------------------
LARGEENVELOPEPOOLSIZE 50 16384
SSLSERVERID * TIMEOUT 30
SSLLIMITS MAXSESSIONS 3000 MAXPERSSLSERVER 600
; -----------------------------------------------------------
   ...
; -----------------------------------------------------------
INFORM
  OPERATOR TCPMAINT MAINT MAINT720
ENDINFORM
; -----------------------------------------------------------
PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
...
  992  TCP INTCLIEN SECURE ITSOSSIA ; Telnet Server (Secure)
  2049 UDP VMNFS               ; NFS Server
  2049 TCP VMNFS     NOAUTOLOG ; NFS Server
; -----------------------------------------------------------
INTERNALCLIENTPARMS
  SECURECONNECTION REQUIRED
  TLSLABEL ITSOSSIA   ; TLS CERT LABEL OF 8 NUM / UPCASE CHAR MAX
   ; INACTIVE 1200    ; CLOSE INACTIVE CONN AFTER 20 MIN IDLE (OPTIONAL)
     TIMEMARK 0600    ; TIMEMARK (KEEPALIVE) CHECK EVERY 10 MIN
     PORT 992         ; ACCEPT SECURE CONN ON TCP/992 (RFC6335)
   ; PORT 23          ; ACCEPT SECURE CONN ON TCP/23 ALSO (OPTIONAL)
ENDINTERNALCLIENTPARMS
; -----------------------------------------------------------
...
```

5. Save your changes with the **FILE** subcommand:

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

6. Use the **QUERY ENROLL ADMIN** command to verify that the TCP/IP installation and service user ID - 6VMTCP30 and both MAINT and MAINT720 are correctly listed as administrators for the VMSYS file pool:

   ```
   ===> query enroll admin for all vmsys
   Number Of Administrators =       8
   MAINT
   MAINT720
   MIGMAINT
   6VMTCP30
   VSMGUARD
   VSMWORK1
   VSMWORK2
   VSMWORK3
   ```

7. Log off TCPMAINT.
8. Log on as **MAINT720**.

9. Ensure that both `MAINT` and `TCPMAINT` are logged off. If either one is logged on, log them off and IPL CMS again before you proceed:

```
===> query maint # query tcpmaint
HCPCQU045E MAINT not logged on
HCPCQU045E TCPMAINT not logged on
```

10. Use VMLINK to access the TCPMAINT 591 and 592 minidisks:

```
===> VMLINK TCPMAINT 592
===> VMLINK TCPMAINT 591
```

11. Run the SSLPOOL utility with the PLAN option to generate an installation plan for the SSL worker pool. When you are prompted to continue, type the numeral `1`, then press **Enter**:

```
===> SSLPOOL PLAN
DTCSLP3372I The SSLPOOL processing mode and values cited here will be used
DTCSLP3397I Processing mode ...........: PLAN
            Options in effect..........:
DTCSLP3396I Operands in effect:
            SFS file pool name ......: VMSYS
            SFS file space owner ID .: TCPMAINT
            SSL server pool prefix ..: SSL
            TCP/IP server ID ........: TCPIP
            SSL server pool count ...: 5
            SSL server work directory: VMSYS:TCPMAINT.SSLPOOL_SSL
            SSL DCSS agent server....: SSLDCSSM
DTCSLP3399R Continue with action PLAN?
            Enter 0 (No), 1 (Yes), 2 (Exit)
===> 1

DTCSLP3381I Creating file SSLPOOL PLANINFO A
DTCSLP3021I SSLPOOL processing completed with RC = 0
```

12. As indicated during processing, the planning output created a new file on the A disk that is called SSLPOOL PLANINFO A. This file contains information that is used to create the updated PROFILE TCPIP for the system by completing the following steps:

    a. Duplicate the file with a new file type of DTCTEMP:

    ```
    ===> copy SSLPOOL PLANINFO A = DTCTEMP A (olddate
    ```

    b. Edit the new file:

    ```
    ===> xedit SSLPOOL DTCTEMP A
    ```

    c. Jump to the first line that begins with `* -----------------`, which is line 15 in Example 6-27. In the prefix area for that line, type **dd** and press **Enter**. The **dd** turns red.

    Example 6-27 Top of SSLPOOL DTCTEMP

    ```
    00000 * * * Top of File * * *
    00001 * =================================================================
    00002 * SSLPOOL PLANINFO         --  SSL Server Pool Planning Info...
    00003 * Created by: SSLPOOL EXEC  --  18 April 2015 - 18:41:15
    ...
    00014 * =================================================================
    dd    * -----------------------------------------------------------------
    00016 * Example SSL Server Pool CP Directory Entry
    ```

d. Jump to the first line that contains the string `nick.SSL` by using the search string. Move to the first blank line above this string and type **dd** in the prefix area for that line. In our example environment, this line was eight lines up at line 80, as shown in Example 6-28:

```
====> /Example DTCPARMS Pool
====> up 8
```

Example 6-28 Example DTCPARMS Pool SSL 'Server' Entry in SSLPOOL DTCTEMP

```
00078 * file (such as ENDVM363 DTCPARMS), or in a SYSTEM DTCPARMS file
00079 * ------------------------------------------------------------------
dd 80
00081 * ==============================================================
00082 * Secure Socket Layer (SSL) - 'SSL' POOL server definition
00083 * > The included :stack. tag identifies the TCP/IP server with which
00084 *   this server pool is associated.
00085 * > The included :vmlink. tag identifies the (common) SFS directory
00086 *   that is to be accessed at file mode A by each pool server
00087 * -------------------------------------------------------------
00088 :nick.SSL*        :type.server   :class.ssl
```

e. Press **Enter** to block-delete the lines between the two sets of **dd**.

f. Jump to the line that contains the string PROFILE TCPIP and move up to the first blank line above it, which is line 57 in Example 6-29. Enter **dd** in the prefix area for this line. Enter another **dd** on the last line of the file, and then press **Enter**:

g. ====> **/PROFILE TCPIP**

h. ====> **up 2**

Example 6-29 Example PROFILE TCPIP

```
dd
00058 * -----------------------------------------------------------------
00059 * Example TCP/IP Server Configuration (PROFILE TCPIP) Modifications
00060 * -----------------------------------------------------------------
00061
...
00073
dd 74 * * * End of File * * *
```

i. Filter to show all lines that contain an asterisk (*):

```
====> all/*
```

j. Suppress any visible lines that are not comment lines by typing an **X** into the prefix area on those two lines, as shown in Example 6-30.

Example 6-30 Suppress lines

```
X 088 :nick.SSL*        :type.server   :class.ssl
...
X 099 :for.SSL*
```

k. Modify the beginning of each comment line to use the correct syntax of **.*** (period asterisk), shift indented lines to the left, and replace parentheses with brackets:

```
====> c/*/.*/* 1
DMSXCG517I 32 occurrence(s) changed on 32 line(s)
====> c/  .*/.*/* *
DMSXCG517I 18 occurrence(s) changed on 18 line(s)
====> c/(/[/* *
====> c/)/]/* *
```

l. Clear the filters. Save your changes. Quit XEDIT by using the `FILE` subcommand:

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

13. Copy the new file to the TCPMAINT 198 disk by completing the following steps:

a. Use VMLINK to access TCPMAINT 198 as file mode U read/write and display the contents by using FILELIST, as shown in Figure 6-14:

```
===> vmlink tcpmaint 198 < 1198 U MR > (filelist
```

```
 MAINT720 FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks   Date       Time
    SYSTEM   DTCPARMS U1 V         71       7      1 2015-04-12 11:04:23
    PROFILE  TCPIP    U1 V         72      78      1 2015-04-18 16:33:12
    PROFILE  TCPIWRKS U1 V         72      61      1 2020-09-19 14:19:19
    PROFILE  TCPIORIG U1 V         73      57      1 2015-04-08 15:45:24
```

Figure 6-14 Initial view of FILELIST

b. Make a backup copy of the existing `SYSTEM DTCPARMS` file by moving your cursor to the beginning of the `SYSTEM DTCPARMS U1` line and typing the following text. You will type over part of the existing text as shown in Figure 6-15. If you make a mistake while you are typing, press **F2** and start over; do *not* use backspace or delete:

```
COPY / = DTCPWRKS = (OLDDATE
```

```
 MAINT720 FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks   Date       Time
COPY / = DTCPWRKS = (OLDDATE       71       7      1 2015-04-12 11:04:23
    PROFILE  TCPIP    U1 V         72      78      1 2015-04-18 16:33:12
    PROFILE  TCPIWRKS U1 V         72      61      1 2020-09-19 14:19:19
    PROFILE  TCPIORIG U1 V         73      57      1 2015-04-08 15:45:24
```

Figure 6-15 Input of data into FILELIST (typing over part of the existing information)

c. After you finish typing, press **Enter** and then, press **F2** to refresh the display.

d. You now see the newly copied file among the other files, as shown in Figure 6-16.

```
 MAINT720 FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks   Date       Time
    SYSTEM   DTCPARMS U1 V         71       7      1 2015-04-12 11:04:23
    SYSTEM   DTCPWRKS U1 V         71       7      1 2015-04-12 11:04:23
    PROFILE  TCPIP    U1 V         72      78      1 2015-04-18 16:33:12
    PROFILE  TCPIWRKS U1 V         72      61      1 2020-09-19 14:19:19
    PROFILE  TCPIORIG U1 V         73      57      1 2015-04-08 15:45:24
```

Figure 6-16 Refreshed FILELIST output that shows newly copied SYSTEM DTCPWRKS

e. Move to the command line at the bottom of the FILELIST panel by pressing **F12**.

f. Populate `SYSTEM DTCPARMS U` with the contents from `SSLPOOL DTCTEMP A`:

i. Move your cursor to the **SYSTEM DTCPARMS U1** line and press **F11** to open the file in XEDIT.

ii. Delete all lines in the current file:

```
====> delete 10
DMSXCG501I 7 line(s) deleted
DMSXSU559W Warning: file is empty
00001 * * * End of File * * *
```

iii. Import the contents of SSLPOOL DTCTEMP A and then move to the top of the file and save your changes so far:

```
====> get SSLPOOL DTCTEMP A
====> top
====> save
```

iv. Modify the header so that it reflects the file name, purpose, and last person to modify the file, as shown in Example 6-31.

Example 6-31 Modified header

```
* =================================================================
* SYSTEM DTCPARMS : TCPMAINT 198 -- MOD 2020-09-19   PWNOVAK
* Created by: SSLPOOL EXEC  --  18 April 2015 - 18:41:15
* =================================================================
```

v. Filter by lines that contain an asterisk (*):

```
====> all/*
```

vi. Block-delete the lines that contain comments that state that they are examples and where to implement them, as shown in Example 6-32.

Example 6-32 Block-delete comments about examples and where to implement them

```
00023 --------------------  3  line(s) not displayed ---------
DD 26 * -----------------------------------------------------
00027 * Example SSL DCSS Management Agent DTCPARMS 'Server' En
00028 * -----------------------------------------------------
00029 * Note: The entries that follow must be implemented with
00030 *       file (such as RDBKZVMH DTCPARMS), or in a SYSTEM
DD 31 * -----------------------------------------------------
00032 --------------------  1  line(s) not displayed ---------
00033   * ===================================================
```

```
00034   * SSL Discontiguous Saved Segment (DCSS) Management Ag
00035   * > The included :stack. tag identifies the TCP/IP ser
00036   *   this server pool is associated.
00037   * ---------------------------------------------------
00038 --------------------  2  line(s) not displayed ---------
00040      :for.SSL*
00041 --------------------  1  line(s) not displayed ---------
DD 42 * -----------------------------------------------------
00043 * Example TCP/IP 'Stack' DTCPARMS 'Server' Entry
00044 * -----------------------------------------------------
00045 * Note: The entries that follow must be implemented with
00046 *       file (such as RDBKZVMH DTCPARMS), or in a SYSTEM
DD 47 * -----------------------------------------------------
00048 --------------------  1  line(s) not displayed ---------
```

vii. Press **Enter** to delete the lines.

viii.Clear the filter, save your changes, and quit XEDIT to return to FILELIST:

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

g. Press **F2** to refresh the display. You see that the record (line) count and date and time stamp differ between SYSTEM DTCPWRKS and SYSTEM DTCPARMS.

h. Press **F3** to quit FILELIST. VMLINK will automatically release and detach the TCPMAINT 198 disk for you and return you to the CMS `Ready;` prompt:

```
DMSVML206I TCPMAINT 198 detached
Ready;
```

14.Test enrollment by using the **`SSLPOOL ENROLL`** command with the **`TEST`** option. It ends with a return code of 4 (`RC = 4`):

```
===> SSLPOOL ENROLL (TEST

DTCSLP3372I The SSLPOOL processing mode and values cited here will be used
DTCSLP3397I Processing mode ...........: ENROLL
            Options in effect..........: TEST
DTCSLP3396I Operands in effect:
            SFS file pool name ......: VMSYS
            SFS file space owner ID .: TCPMAINT
            SSL server pool prefix ..: SSL
            TCP/IP server ID ........: TCPIP
            SSL server pool count ...: 5
            SSL server work directory: VMSYS:TCPMAINT.SSLPOOL_SSL
            SSL DCSS agent server....: SSLDCSSM
DTCSLP3399R Continue with action ENROLL?
            Enter 0 (No), 1 (Yes), 2 (Exit)

===> 1

DTCSLP3360W Option TEST is in effect; Commands prefaced with '*:>' not employed
...
DTCSLP3021W SSLPOOL processing completed with RC = 4
Ready(00004);
```

15. Run the enrollment by using the **SSLPOOL ENROLL** command. It ends with a return code of 0 (RC = 0):

```
===> SSLPOOL ENROLL
DTCSLP3371I User ID MAINT720 administrative authority confirmed for file pool
VMSYS
DTCSLP3374I Checking VMSYS enrollment status of user ID TCPMAINT
DTCSLP3377I User ID TCPMAINT is enrolled in filepool VMSYS
DTCSLP3375I Checking VMSYS storage limits for user ID TCPMAINT
DTCSLP3379I Creating 'SSL' server pool 'work' directory:
VMSYS:TCPMAINT.SSLPOOL_SSL
DTCSLP3388I Processing user ID SSL00001
DTCSLP3374I Checking VMSYS enrollment status of user ID SSL00001
DTCSLP3377I User ID SSL00001 is enrolled in filepool VMSYS
DTCSLP3388I Processing user ID SSL00002
DTCSLP3374I Checking VMSYS enrollment status of user ID SSL00002
DTCSLP3377I User ID SSL00002 is enrolled in filepool VMSYS
DTCSLP3388I Processing user ID SSL00003
DTCSLP3374I Checking VMSYS enrollment status of user ID SSL00003
DTCSLP3377I User ID SSL00003 is enrolled in filepool VMSYS
DTCSLP3388I Processing user ID SSL00004
DTCSLP3374I Checking VMSYS enrollment status of user ID SSL00004
DTCSLP3377I User ID SSL00004 is enrolled in filepool VMSYS
DTCSLP3388I Processing user ID SSL00005
DTCSLP3374I Checking VMSYS enrollment status of user ID SSL00005
DTCSLP3377I User ID SSL00005 is enrolled in filepool VMSYS
DTCSLP3384I Granting 'work' directory authorizations to server SSL00001
DTCSLP3384I Granting 'work' directory authorizations to server SSL00002
DTCSLP3384I Granting 'work' directory authorizations to server SSL00003
DTCSLP3384I Granting 'work' directory authorizations to server SSL00004
DTCSLP3384I Granting 'work' directory authorizations to server SSL00005
DTCSLP3373I Processing server pool PROFILE EXEC file (CREATE)
DTCSLP3378I Creating server pool PROFILE EXEC (from file TCPROFIL EXEC *)
DTCSLP3373I Processing server pool PROFILE EXEC file (SETALIAS)
DTCSLP3383I Establishing server pool alias to common-use PROFILE EXEC
DTCSLP3380I Creating alias for server SSL00001
DTCSLP3380I Creating alias for server SSL00002
DTCSLP3380I Creating alias for server SSL00003
DTCSLP3380I Creating alias for server SSL00004
DTCSLP3380I Creating alias for server SSL00005
DTCSLP3021I SSLPOOL processing completed with RC = 0
```

16. Run the **SSLPOOL** command with the **SETAUTH** option to set the correct authorizations:

```
===> SSLPOOL SETAUTH
DTCSLP3372I The SSLPOOL processing mode and values cited here will be used
DTCSLP3397I Processing mode ...........: SETAUTH
            Options in effect..........:
DTCSLP3396I Operands in effect:
            SFS file pool name ......: VMSYS
            SFS file space owner ID .: TCPMAINT
            SSL server pool prefix ..: SSL
            Administrative ID .......: TCPMAINT
            SSL server work directory: VMSYS:TCPMAINT.SSLPOOL_SSL
DTCSLP3399R Continue with action SETAUTH?
            Enter 0 (No), 1 (Yes), 2 (Exit)
```

```
===> 1

DTCSLP3371I User ID MAINT720 administrative authority confirmed for file pool
VMSYS
DTCSLP3374I Checking VMSYS enrollment status of user ID TCPMAINT
DTCSLP3377I User ID TCPMAINT is enrolled in filepool VMSYS
DTCSLP3384I Granting 'work' directory authorizations to user TCPMAINT
DTCSLP3021I SSLPOOL processing completed with RC = 0
```

17. Erase the temporary file that is used to hold the parms value. Then, log off as MAINT720:

```
===> erase SSLPOOL DTCTEMP A
===> logoff hold
```

18. Log on as the **GSKADMIN** user ID and allow its default PROFILE EXEC to run. You see the following information:

```
LOGON GSKADMIN
z/VM Version 6 Release 3.0, Service Level 1501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
...
Profile..: Spooling console to self (GSKADMIN)...
Profile..: Setting PF Keys...
PF12 RETRIEVE BACKWARD
PF24 RETRIEVE BACKWARD
Profile..: Setting minidisk environment workspace...
DMSACC724I 191 replaces A (191)
DMSACP723I E (591) R/O
DMSACP723I F (592) R/O
Profile..: Setting up BFS environment...
Profile..: Determining what is currently mounted...
Nothing is mounted
Profile..: Mounting root file system...
Profile..: Mounting GSKSSLDB file space at: /etc/gskadm/
Profile..: Setting working directory to: /etc/gskadm/
Profile..: (for direct access to key database files)...
Profile..: Checking mounts...
Mount point = '/etc/gskadm'
Type  Stat  Mounted
BFS   R/W  '/../VMBFS:VMSYS:GSKSSLDB/'
Mount point = '/'
Type  Stat  Mounted
BFS   R/W  '/../VMBFS:VMSYS:ROOT/'

Profile..: Checking current directory content...
DMSOVK1229E /etc/gskadm is empty

Profile..: Setup complete; Environment prepared for use of GSKKYMAN
Ready;
```

19. Clear your panel and then, run the **GSKKYMAN** utility. At the top-level menu, select **Create new database** and respond to the prompts, as shown in Figure 6-17 on page 231.

```
===> vmfclear
===> gskkyman
```

Important: As you begin working with GSKit utilities, be aware that they use the OpenVM Bit File System, which means that you are now effectively working on a UNIX or an AIX system. Commands, file names, paths, and passwords are case-sensitive.

20. Select **option 1** to create a key database, and use the default file name of **`Database.kdb`** (note the upper case D).
21. Select a password to use for securing the database. This password can be as complex as you want, but remember the following information:
    - Remember that the password is case-sensitive.
    - The number sign (#) character is the end of line indicator; do not use it in your password.
    - Your password must not be trivial.
    - Document the password in a secure location, such as an enterprise identity and access management (IAM) data vault.
22. If a requirement or regulation mandates that cryptography-stored passwords must change on a timed interval, set an expiration date. If not, press **Enter**.
23. Press **Enter** twice to set the record length to the default of 5,000.
24. If you are required to abide by United States Government Federal Information Processing Standards (FIPS), answer **`1`** to the FIPS prompt. The typical answer is `0`, but check with your business controls office if you are unsure.
25. After you are notified that the key database is created, press **Enter** twice to return to the top-level menu.

```
 Database Menu

   1 - Create new database
...

Enter option number:
1
Enter key database name (press ENTER to return to menu):
Database.kdb
Enter database password (press ENTER to return to menu):
P@ssw0rd4zVMgsk!
Re-enter database password:
P@ssw0rd4zVMgsk!
Enter password expiration in days (press ENTER for no expiration):
Enter Enter
Enter database record length (press ENTER to use 5000):
Enter Enter
Enter 1 for FIPS mode database or 0 to continue:
0
Key database /etc/gskadm/Database.kdb created.

Press ENTER to continue.
Enter Enter
```

Figure 6-17 Initial run of GSKKYMAN

26.Select option **10** to stash the password that you set into an encrypted file. The SSL-TLS server will use this stash file during run time to access the key database.

27.After you are notified that the stash file is created, press **Enter** twice to return to the top-level menu. Then, select option **0** to exit from the utility.

28.Issue the following **OPENVM** commands to ensure that the necessary database files were created and to list the permissions of these files:

```
===> openvm list /etc/gskadm/
Directory = '/etc/gskadm/'
Update-Dt  Update-Tm Type  Links          Bytes Path name component
04/28/2015 13:32:14   F        1         105080 'Database.kdb'
04/28/2015 13:42:54   F        1             80 'Database.rdb'
04/28/2015 13:33:06   F        1            129 'Database.sth'

===> openvm list /etc/gskadm/ (own
Directory = '/etc/gskadm/'
User ID    Group Name  Permissions Type  Path name component
gskadmin   security    rw- --- ---  F    'Database.kdb'
gskadmin   security    rw- --- ---  F    'Database.rdb'
gskadmin   security    rw- --- ---  F    'Database.sth'
```

29.Issue the following **OPENVM PERMIT** commands to allow the SSL-TLS server to access the new key database:

```
===> openvm permit /etc/gskadm/Database.kdb rw- r-- ---
===> openvm permit /etc/gskadm/Database.rdb rw- r-- ---
===> openvm permit /etc/gskadm/Database.sth rw- r-- ---
```

30.Confirm that r (read) was added to the “group” permissions for the key database and password stash files:

```
===> openvm list /etc/gskadm/ (own
Directory = '/etc/gskadm/'
User ID    Group Name  Permissions Type  Path name component
gskadmin   security    rw- r-- ---  F    'Database.kdb'
gskadmin   security    rw- --- ---  F    'Database.rdb'
gskadmin   security    rw- r-- ---  F    'Database.sth'
```

31.With the key database now in place, you can start the SSL server to confirm that it can access this database; but first, we generate a self-signed certificate for testing.

> **Note:** Do not attempt to log on to the SSL server through a secure Telnet connection. An attempt to log on to the SSL server through a secure Telnet connection will be rejected with the message:
>
> `HCPLGA206E Cannot connect to host virtual machine`
>
> For more information, see “TCP/IP and SSL Server Logon Restrictions”, in *z/VM TCP/IP Planning and Customization*, SC24-6125.

32. Create a self-signed certificate for testing purposes by completing the following steps:

> **Important:** The use of self-signed certificates is not recommended for production environments. Use self-signed certificates only in test environments before production.

a. Clear your panel, then run the **GSKKYMAN** utility. At the top-level menu, select option **2 - Open** database, enter **Database.kdb**, and then, enter the password in response to the prompts:

```
===> vmfclear
===> gskkyman
===> 2
===> Database.kdb
===> P@ssw0rd4zVMgsk!
```

b. The Key Management Menu appears. Select option **6 - Create a self-signed certificate**:

```
===> 6
```

c. At the Select certificate type prompt, choose option **6 - User or server certificate with 2048-bit RSA key**:

```
===> 6
```

d. At the Select digest type prompt, choose option **3 - SHA-256**:

```
===> 3
```

e. Respond to the label and subject name prompts by using Figure 6-18 on page 234 as a guide. Replace the example values with those values that are correct for your environment. The value for the label must match the value that was used in step 3 on page 222.

```
Enter label (press ENTER to return to menu):
ITSOSSIA
Enter subject name for certificate
  Common name (required):
*.itso.ibm.com

  Organizational unit (optional):
ITSO Redbooks SG248147

  Organization (required):
International Business Machines Corporation

  City/Locality (optional):
Endicott

  State/Province (optional):
New York

  Country/Region (2 characters - required):
US

Enter number of days certificate will be valid (default 365):
1460

Enter 1 to specify subject alternate names or 0 to continue:
0

Please wait .....
Certificate created.
Press ENTER to continue.
```

Figure 6-18 Example values for self-signed certificate subject name fields

f. Press **Enter** twice to return to the Key Management Menu and select the following options:

```
===> 1 - Manage keys and certificates
===> 1 - ITSOSSIA
```

(the option that displays the label of the certificate that you generated)

```
===> 3 - Set key as default
```

g. Press **Enter** twice to return to the Key and Certificate Menu and then, select **0** to exit the GSKKYMAN utility.

33. Grant the LOGONBY privilege for GSKADMIN to the authorized IDs. Then, change the password for GSKADMIN to LBYONLY:

```
===> dirmaint for GSKADMIN logonby add spiedie tjwatson lydiap pwnovak
Your LOGONBY request for GSKADMIN at * has completed; with RC = 0.
===> dirmaint for GSKADMIN SETPW LBYONLY
DVHXMT1191I Your SETPW request has been sent for processing to DIRMAINT ...
DVHREQ2289I Your SETPW request for GSKADMIN at * has completed; with RC = 0.
```

34. Repeat step 32 for TCPMAINT and 6VMTCP30.

35. Repeat the previous steps on all other members of the SSI cluster.

From this point forward, you must ensure that your 3270 emulator is correctly configured to connect by using an encrypted connection. An example of this connection from a Linux or Mac workstation uses a command, such as the following command from the terminal:

```
$ x3270 -accepthostname any L:endvm363.wsclab.endicott.ibm.com:992
```

Use the help documentation for your 3270 emulator to obtain the correct details that you will require to configure your 3270 emulator to connect by using an encrypted connection.

6.15 Backing up and restoring your z/VM system

Your SSI system is now customized with running TCP/IP stacks, two highly available virtual switches, a start-up and shutdown process, TLS encryption, and shared CMS utilities in the common SFS file pool. You changed the passwords. Now is a good time to back up the system to tape. See Appendix E, "Back up the z/VM system to tape" in the *IBM z/VM V7R2 Installation Guide,* GC24-6246.

Backing up and restoring data are essential components of data storage management. Backing up your data on a regular basis helps protect your system against the loss of data if a major disaster occurs, or when data is accidentally deleted or becomes corrupted.

Planning for full system and incremental backup capabilities is an integral part of the migration plan for many customers who are migrating workloads to Linux on IBM Z. Depending on the configuration of the IBM Z environment, customers can select to use a z/VM specific strategy, Linux specific strategy, or a combination of the two for their backup plan.

z/VM centric solutions can provide the following types of backups:

- File level backup and recovery of CMS data

  This type is useful when you need to recover a file (or small number of files) because of administrative or operational errors. Having file level backups of the z/VM hypervisor can be critical in reducing or preventing an outage associated with the hypervisor, which affects the availability of all guest systems.

- Image level backup and recovery of z/VM systems

  This type is useful when you need to recover an entire minidisk or an entire DASD volume because of administrative or operational errors, hardware issues, or disaster recovery situations.

- Image level backup and recovery of Linux guests

  These backups can provide for faster recovery if a major failure of a Linux guest or system-wide disaster occurs. For example, you must restore an entire minidisk that is owned by a Linux guest, an entire guest, or one or more DASD volumes that are owned by a Linux guest.

Linux-centric solutions can provide file level backup and recovery for Linux guest systems; that is, the solution or its client is running in Linux and understands the file system. Again, this function is useful when you need to recover a file (or small number of files) because of administrative or operational errors.

By including a combined z/VM and Linux solution in your backup strategy, you can support:

- File level backup and recovery of:
  - z/VM data
  - Linux data
- Image level backup and recovery of z/VM and Linux systems

By combining Backup and Restore Manager for z/VM and TSM, a comprehensive backup and recovery solution for the z/VM and Linux on IBM Z can be provided. Disaster-level backup and recovery is available for z/VM and the Linux guests.

The Linux virtual machines are backed up as part of the z/VM system on which they are hosted. Equally, the TSM client code is backed up with the Linux virtual machines, which allows the TSM solution to perform its needed functions when the Linux virtual machine is again operational. In addition, file level backup and recovery also is available for z/VM and the Linux guests, which provides easier recovery from operational errors.

IBM offers different and sometimes complementary strategies, such as IBM Backup and Restore Manager for z/VM and IBM Tivoli® Storage Manager (TSM) for Linux (see Figure 6-19) that can be part of your backup and restore master plan.

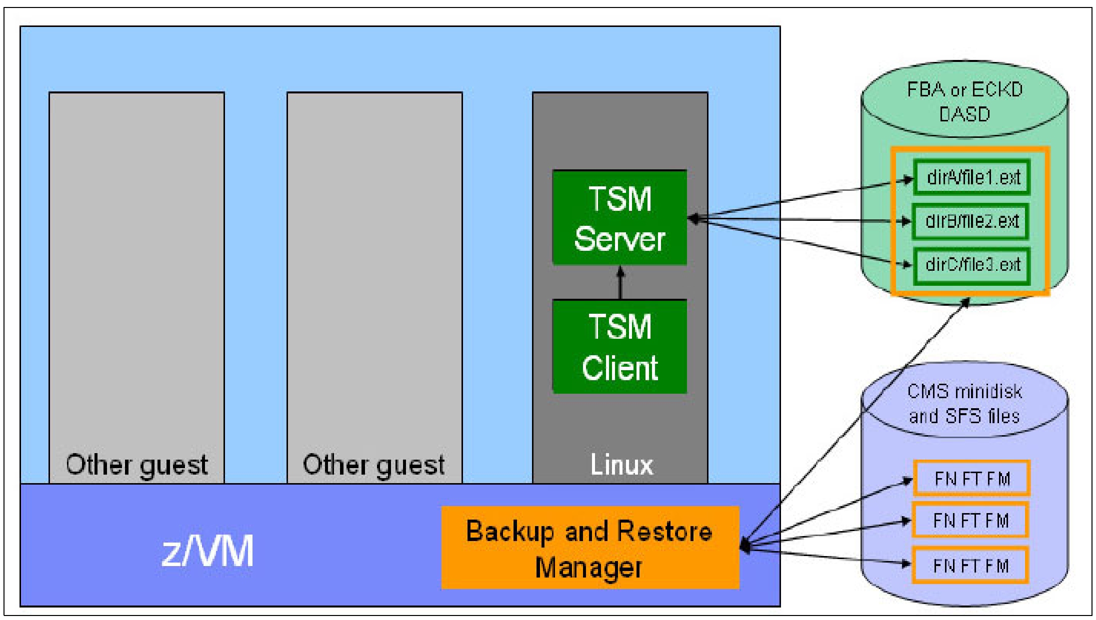


Figure 6-19 Using Backup and Restore Manager with Tivoli Storage Manager.

> If you are interested in having a product to backup your system, such as IBM Backup Restore, see Chapter 8 IBM Backup and Restore Manager for z/VM of *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039.

Practice restoring a system

You do not want your first restore to be the result of an emergency. After you complete the backup, attempt to restore your system as described on Appendix H, “Restore the z/VM system backup from tape” in IBM z/VM V7R2 Installation Guide, GC24-6292.

If you do not have a tape device, appendixes exist that describe backing up and restoring to and from DASD. For more information about backup solutions, see this web page.

6.16 Creating an SFS file pool for Linux virtual machines

Within z/VM, several options are available to provide a 191 disk to each of the users. Common implementations include sharing a read-only minidisk that is owned by LNXMAINT, or defining a real minidisk for each of the z/VM guest systems.

In this book, the authors implemented a sophisticated way to provide each user with a writable 191 disk and still share part of the content among all of the guests. In z/VM, the chosen solution is called a Shared File System (SFS) file pool.

Each file pool is a collection of minidisks that are owned by a particular file pool virtual machine, which is known as a *file pool server*. The minidisks are used for storing file pool repository data, with control data (for example, catalogs, logs, and parameter files) that is necessary to keep the data definitions and recovery information.

6.16.1 SFS file pools characteristics

When you compare the SFS file pool with technologies that are available to Linux, compare it with NFS.

> **Note:** By default, SFS file pools are not enabled for remote access, but remote access can be enabled, if you want. You can set up a connection by using the Transparent Services Access Facility (TSAF), the Inter System Facility for Communication (ISFC), or Advanced Program-to-Program Communication (APPC)/VM Virtual Telecommunications Access Method (IBM VTAM®) Support (AVS). For more information, see “Setting Up a File Pool for Remote Use” in *IBM z/VM CP Planning and Administration*, SC24-6178.

The following information relates to SFS file pools:

- SFS file pools are run by a specific z/VM guest and need a special user with the appropriate rights to configure them; in our example environment, the users are LNXSERV1 and LNXMAINT.
- SFS file pools are structured in a tree style and they can contain subdirectories.
- Each user is enrolled (through ENROLL) to be granted access and allocated a certain amount of disk space, similar to Linux user quotas. File pools can be used to provide the equivalent of a 191 (A) disk to z/VM guests.
- File pools can share part of the content among z/VM guests.
- SFS file pools have more granular access controls. They also can ALIAS a file, which is similar to a Linux symbolic link.
- File pools are accessible through CMS only. They are not used as a base for Linux volumes nor are they accessible to Linux by using fusemount.
- For file pool IDs without a VMSYS prefix, SSI indicates that the server must accept a file pool ID connection from outside the processor on which it is running, but only if the request is from another member of the same SSI cluster.
- It is critical that you frequently back up the SFS file pool.

6.16.2 Adding a directory entry for the new SFS server machine

Our new SFS file pool for use by Linux virtual machines will run under a dedicated VSM that is named LNXSERV1.

Follow these steps:

1. As MAINT or MAINT720, use XEDIT to create a directory entry for user LNXSERV1:

   ```
   ===> xedit LNXSERV1 DIRECT A
   ```

2. Use the entries in the example directory entry that is shown in Example 6-33 to populate the file. On the planning worksheet, a volume was specifically designated as the target for this SFS server's minidisks in 2.13.5, "z/VM DASD" on page 67. That volume is VM156A in our environment.

   MDISK 0191 is too small to run the backup later. Increase MDISK 0191 to at least 10 cylinders. (More cylinders might be required during normal operation.)

   Example 6-33 Sample directory entry for a new SFS server machine

   ```
   USER LNXSERV1 LBYONLY 64M 128M BG
    INCLUDE IBMDFLT
      LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNOVAK SPIEDIE
      ACCOUNT 1 LNXMAINT
      IPL CMS
      IUCV ALLOW
      IUCV *IDENT RESANY GLOBAL
      MACH XC
      OPTION MAXCONN 2000 NOMDCFS APPLMON ACCT QUICKDSP SVMSTAT
      POSIXOPT SETIDS ALLOW
      SHARE RELATIVE 1500
      XCONFIG ADDRSPACE MAXNUMBER 100 TOTSIZE 8192G SHARE
      XCONFIG ACCESSLIST ALSIZE 1022
      CONSOLE 0009 3215 T OPERATOR
      LINK MAINT 0190 0190 RR
      LINK MAINT 0193 0193 RR
      LINK MAINT 019D 019D RR
      MDISK 0091 3390 0001 0001 VM156A RR
      MDISK 0191 3390 0202 0001 VM156A MR
      MDISK 0192 3390 0203 0100 VM156A MR
      MDISK 0301 3390 0303 0009 VM156A WR
       MINIOPT NOMDC
      MDISK 0302 3390 0312 0014 VM156A WR
       MINIOPT NOMDC
      MDISK 0303 3390 0326 0014 VM156A WR
       MINIOPT NOMDC
      MDISK 0304 3390 0340 0015 VM156A WR
      MDISK 0305 3390 0355 0500 VM156A WR
   ```

3. After you add all of the lines, issue the subcommand FILE to save the changes and quit XEDIT:

   ```
   ====> FILE
   ```

4. Send the directory entry to DirMaint for processing:

   ```
   ===> dirm add lnxserv1
   ```

5. Enable the automatic logon of the ID during system start-up:

```
===> dirmaint for autolog1 xautolog add lnxserv1
...
DVHREQ2289I Your XAUTOLOG request for AUTOLOG1 at * has completed; with RC=0
```

Allow several minutes for any DirMaint asynchronous processing of LNXSERV1 to complete.

6.16.3 Generating the SFS file pool for Linux guest systems

Now that LNXSERV1 is built, generate the file pool that will run under it. This file pool is named LNX. Complete the following steps:

1. Log on to **LNXSERV1** on the first member. CMS automatically loads, but it displays an error message:

   ```
   A(191) device error
   ```

   The error occurs because the 191 minidisk is not formatted yet. Resolve this error by formatting and labeling the 191 minidisk:

   ```
   ===> format 191 A
   Erase all files?
   ===> 1
   Enter label:
   ===> LNX191
   ```

2. Create a PROFILE EXEC for this server machine, as shown in Figure 6-20:

   ```
   ===> xedit profile exec a
   ```

   ```
   /*** LNXSERV1 PROFILE EXEC : LNXSERV1 191 -- MOD 2015-04-10 PWNOVAK ***/
   ADDRESS COMMAND
   'CP MSG OPERATOR LOGON 'USERID()' FOR LINUX SFS FILEPOOL LNX:'
   'CP SPOOL CONSOLE START'
   'CP SPOOL CONSOLE TO OPERATOR EOF'
   'CP SET PF11 RETRIEVE BACK'
   'CP SET PF12 RETRIEVE'
   'ACCESS 193 C'
   'CP SET EMSG ON'
   'CP SET RUN ON'
   'SET AUTOREAD OFF'
   'EXEC FILESERV START'
   IF DISC() THEN
     'CP LOGOFF'
   EXIT

   DISC: RETURN (SUBSTR(DIAG(24,-1),13,1)<>0)
   ```

   Figure 6-20 LNXSERV1 PROFILE EXEC contents

   The ACCESS command is necessary because several of the files that are needed by the server on MAINT's 193 minidisk. Specify SET EMSG ON so that the message number will be included in the messages that are shown on the server. You can use the message number to look up messages in *z/VM: CMS and REXX/VM Messages and Codes,* GC24-6118.

3. After you create the PROFILE EXEC, file it, then copy it as SETUP EXEC:

   ```
   ====> FILE
   ```

```
===> copy profile exec a setup = =
```

4. Edit SETUP EXEC and delete these lines:

```
'CP SET RUN ON'
'SET AUTOREAD OFF'
'EXEC FILESERV START'
```

5. Create a start-up parameters file, as shown in Figure 6-21:

```
===> xedit lnxserv1 dmsparms a
```

```
ADMIN LNXMAINT MAINT MAINT720 FTPSERVE
BACKUP
FILEPOOLID LNX
SAVESEGID CMSFILES
REMOTE
USERS 300
NOSHUTDOWNSIGNAL
NODFSMS
```

Figure 6-21 LNXSERV1 DMSPARMS contents

6. Issue the following commands to generate the file pool:

```
===> access 193 c
===> fileserv generate
DMS4PD3400I Initializing begins for DDNAME = CONTROL
DMS4PD3400I Initializing ends for DDNAME = CONTROL
DMS4PD3400I Initializing begins for DDNAME = MDK00001
DMS4PD3400I Initializing ends for DDNAME = MDK00001
DMS4PD3400I Initializing begins for DDNAME = MDK00002
DMS4PD3400I Initializing ends for DDNAME = MDK00002
DMS4PD3400I Initializing begins for DDNAME = LOG1
DMS4PD3400I Initializing ends for DDNAME = LOG1
DMS4PD3400I Initializing begins for DDNAME = LOG2
DMS4PD3400I Initializing ends for DDNAME = LOG2
...
```

FILESERV GENERATE performs the following functions:

– FILESERV GENERATE issues CMS FORMAT and RESERVE commands for the file pool minidisks. Depending on the number and size of your initial minidisks, this process takes a long time.

– FILESERV GENERATE initializes the file pool minidisks.

– FILESERV GENERATE processing places internal control information on the file pool minidisks. This control information is needed for usual server operation.

– FILESERV GENERATE creates the POOLDEF file. The POOLDEF file has a file name that is the same as the file pool ID that you entered in the LNXSERV1 DMSPARMS file in the last step. The file type is POOLDEF. FILESERV GENERATE processing creates the file on the first read/write file mode in the server machine's search order, which happens to be the 191 work disk.

7. During its processing, FILESERV GENERATE calls XEDIT to display a file that contains control statements. When this step is reached, XEDIT opens and displays the contents of the IBM default values for the POOLDEF file, as shown in Figure 6-22 on page 241.

```
 $$TEMP   $POOLDEF A1  F 80  Trunc=80 Size=10 Line=0 Col=1 Alt=0

00000 * * * Top of File * * *
00001 MAXUSERS=1000
00002 MAXDISKS=500
00003 DDNAME=CONTROL        VDEV=301
00004 DDNAME=LOG1           VDEV=302
00005 DDNAME=LOG2           VDEV=303
00006 DDNAME=BACKUP   DISK  FN=FILEPOOL   FT=BACKUP   FM=*
00007 DDNAME=MDK00001       VDEV=304      GROUP=1     BLOCKS=0
00008 DDNAME=MDK00002       VDEV=305      GROUP=2     BLOCKS=0
D 009 DDNAME=CRR1           VDEV=306
D 010 DDNAME=CRR2           VDEV=307
00011 * * * End of File * * *

====>
                                                      X E D I T  1 File
```

Figure 6-22 POOLDEF defaults with D in the prefix area of lines 9 and 10

8. Modifications are required. Enter **D** into the prefix area on lines 9 and 10 as shown in Figure 6-22 and press **Enter**. After you delete these two lines, save and exit:

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

   FILESERV GENERATE processing continues by using the control statements that you specified:

   ```
   DMS5FD3032I File pool server has terminated
   DMSWFV1120I File LNX POOLDEF A created or replaced
   DMSWFV1117I FILESERV processing ended at 17:18:11 on 10 Apr 2015
   ```

9. Back up the file pool control data:

   ```
   ===> FILESERV BACKUP
   DMSWFV1117I FILESERV processing begun at 17:47:15 on 10 Apr 2015
   DMSWFV1121I LNXSERV1 DMSPARMS A1 will be used for FILESERV processing
   DMSWFV1121I LNX POOLDEF A1 will be used for FILESERV processing
   DMS4HA3239I The DDNAME=BACKUP file is being created with the following
   DMS4HA3239I timestamp: 04-10-15 17:47:15
   DMS4HA3293I 04-10-15 17:47:15 File pool control data backup starting
   DMS4GL3294I 04-10-15 17:47:15 File pool control data backup complete
   DMS5FD3032I File pool server has terminated
   DMSWFV1117I FILESERV processing ended at 17:47:15 on 10 Apr 2015
   ```

10. Start the server to access the file pool in multiple user mode:

    ```
    ===> FILESERV START
    DMSWFV1117I FILESERV processing begun at 17:47:26 on 10 Apr 2015
    DMSWFV1121I LNXSERV1 DMSPARMS A1 will be used for FILESERV processing
    DMSWFV1121I LNX POOLDEF A1 will be used for FILESERV processing
    DMS5BB3045I Ready for operator communications
    ```

11. After you see the “`DMS5BB3045I Ready for operator communications`” message appear, disconnect:

    ```
    ===> #CP DISCO
    ```

6.16.4 Adding a directory entry for the SFS administration machine

The `LNXMAINT` virtual machine now must be created. LNXMAINT owns and maintains the LNX SFS file pool. Complete the following steps:

1. Create the entry and then, send it to DirMaint for processing:

   ```
   ===> xedit lnxmaint direct a
   ```

   You see the information that is shown in Figure 6-23.

   ```
   USER LNXMAINT LBYONLY 32M 128M BG
   INCLUDE IBMDFLT
   IPL CMS PARM FILEPOOL LNX AUTOCR
      LOGONBY AUTOLOG1 BG EDIALVES PARZIALE MSOUZA VIC PWNOVAK
      MACHINE ESA
   ```

   Figure 6-23 Entry created and sent to DirMaint for processing

   ```
   ===> dirmaint add lnxmaint
   ...
   DVHREQ2289I Your ADD request for LNXMAINT at * has completed; with RC=0
   ```

2. Use LOGONBY to log on to LNXMAINT. During IPL, you might initially see errors that state that a directory was not found or is not authorized for access. These error messages are normal during this initial setup, and can be safely ignored:

   ```
   ===> logon lnxmaint by edialves
   LOGON LNXMAINT BY PWNOVAK
   LNXMAINT AT RDBKZVMG VIA *        04/10/15 18:37:54 EDT      FRIDAY
   Ready;
   ```

   Example 6-34 LOGONBY syntax for the z/VM logon panel

   ```
   ...
    Fill in your USERID and PASSWORD and press ENTER
    (Your password will not appear when you type it)
    USERID   ===>
    PASSWORD ===>

    COMMAND  ===>  LOGON LNXMAINT BY PWNOVAK
                                                     RUNNING   RDBKZVMG
   ```

3. Enroll LNXMAINT in the LNX file pool with a limit of 500 4K blocks. This amount is more than sufficient, but you can adjust it in the future, if required:

   ```
   ===> enroll user LNXMAINT lnx ( blocks 500
   ```

4. Access the file pool, which accesses the user directory for LNXMAINT in the LNX file pool:

   ```
   ===> access
   Ready;
   ```

5. Check to ensure that the file pool directory for LNXMAINT appears as file mode A:

   ```
   ===> query accessed
   Mode  Stat     Files  Vdev  Label/Directory
   A      R/W         0  DIR   LNX:LNXMAINT.
   S      R/O       698  190   MNT190
   Y/S    R/O      1123  19E   MNT19E
   Ready;
   ```

6. Create a PROFILE EXEC for this server machine by using the information in Figure 6-24:

```
===> xedit profile exec a
```

```
/*** LNXMAINT PROFILE EXEC : LNX:LNXMAINT. -- MOD 2015-04-10 PWNOVAK ***/
ADDRESS COMMAND
'CP SET PF11 RETRIEVE BACK'
'CP SET PF12 RETRIEVE'
'EXEC VMLINK .DIR LNX:LNXADMIN. < . D FORCERW >'
'CP SET RUN ON'
EXIT
```

Figure 6-24 LNXMAINT PROFILE EXEC contents

7. Enroll LNXADMIN, which holds all of the contents that are used to punch the Linux kernel and start the FILESERV GENERATE. We allocate 30,000 blocks initially, which can be increased later as your environment expands:

```
===> enroll user lnxadmin lnx ( blocks 30000
```

8. Start the new profile:

```
===> profile
```

The virtual machine that is the Linux administrative system is now defined. Remain logged in as LNXMAINT and proceed to perform the initial enrollments.

6.16.5 Enrolling the Linux virtual machines as USERS

While you are still logged in as LNXMAINT, complete the following steps to create the directories, enrollments, and authorizations for the file pool:

1. (Optional) If you run multiple Linux distributions, or think you might do so in the future, create directories to be used for separation of different items:

```
===> create directory lnx:lnxadmin.swapgen
===> create directory lnx:lnxadmin.redhat
===> create directory lnx:lnxadmin.suse
===> create directory lnx:lnxadmin.ubuntu
```

2. Grant public read authorizations for the LNXADMIN directory and any files inside of it.

   PUBLIC means any USER (or IDENTITY) enrolled in the LNX filepool has read-only access to the LNXADMIN directory; all others have no visibility:

```
===> grant auth lnx:lnxadmin. to public ( read newread
```

   If you are granting public authorizations for an existing user directory, all of the files in the directory can be made accessible to enrolled IDs with the following command:

```
===> grant auth * * lnx:lnxadmin. to public ( read
```

> **Note:** For suitable security, the only IDs that are enrolled in the LNX filepool should be Linux virtual servers, admin IDs that are defined in Figure 6-21, and I/T staff supporting z/VM and Linux virtual servers.

   a. If you created directories, grant public read authorizations for them and any files inside:

```
===> grant auth lnx:lnxadmin.swapgen to public ( read newread
===> grant auth lnx:lnxadmin.redhat to public ( read newread
===> grant auth lnx:lnxadmin.suse to public ( read newread
```

```
===> grant auth lnx:lnxadmin.ubuntu to public ( read newread
```

3. Create enrollment of the first few Linux virtual server machines in the file pool with 100 blocks each. Note that these IDs are not required to exist in the directory yet; you are creating configuration that SFS will utilize if and when the ID is encountered:

```
====> enroll user linux1 lnx ( blocks 100
====> enroll user linux2 lnx ( blocks 100
====> enroll user linux3 lnx ( blocks 100
====> enroll user linux4 lnx ( blocks 100
====> enroll user linux5 lnx ( blocks 100
====> enroll user linux6 lnx ( blocks 100
====> enroll user lnxs0001 lnx ( blocks 100
====> enroll user lnxs0002 lnx ( blocks 100
====> enroll user lnxs0003 lnx ( blocks 100
```

6.16.6 Adding Linux parm files and REXX EXECs to the LNX file pool

While you are still logged in as LNXMAINT, complete the following steps:

1. Use VMLINK to access the TCP/IP tools so that you can use the z/VM FTP client:

```
===> vmlink tcpmaint 592
DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
```

2. Ensure that you still have the SFS directory for **LNXADMIN** accessed as file mode **D** with R/W (read/write) status:

```
===> query accessed
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         1  DIR   LNX:LNXMAINT.
D      R/W         1  DIR   LNX:LNXADMIN.
S      R/O       698  190   MNT190
Y/S    R/O      1124  19E   MNT19E
Z      R/O       892  120   TCM592
```

   – If it is not accessed or it is not read/write status, run the command:

```
===> ACCESS LNXADMIN. D (FORCERW
DMSACR724I LNX:LNXADMIN. replaces D
DMSACR724I (LNX:LNXADMIN.)
```

   – If you created directories, access them as well in R/W status:

```
===> ACCESS LNXADMIN.SWAPGEN E (FORCERW
Ready; T=0.01/0.01 16:40:03
===> ACCESS LNXADMIN.REDHAT F (FORCERW
===> ACCESS LNXADMIN.SUSE G (FORCERW
===> ACCESS LNXADMIN.UBUNTU H (FORCERW
```

3. On the FTP server. the directory path `/ftp/zvm/cookbook/lnxmaint/` (or `/ftp/zvm/sg248147/lnxmaint/`) was created automatically through the expansion of the .tgz file in 5.2.1, "Creating directories on the FTP server and upload the installation image" on page 108. We are now ready to transfer files from the FTP server to the LNX file pool using one of the following methods:

   – Perform a **get** (pull) from z/VM by initiating a session to the FTP server:

```
===> ftp 9.60.87.87
...
===> lcd D
```

```
===> cd /ftp/zvm/cookbook/lnxmaint
===> mget *
===> quit
```

– Perform a **put** (push) from the FTP server by initiating a session to z/VM:

```
===> ftp RDBKZVMG.itso.ibm.com
...
===> lcd D
===> cd /ftp/zvm/cookbook/lnxmaint
===> mget *
===> quit
```

4. List the files that you downloaded into SFS. The list shows numerous files:

```
===> vmfclear
===> listfile * * D (isodate
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
GENERIC  PRM      D1 V         66          7          1 2015-04-20 15:28:54
PROFCKD  EXEC     D1 V         72         37          1 2015-04-27 15:11:21
PROFFBA  EXEC     D1 V         72         49          1 2015-04-27 15:11:10
PROFILE  EXEC     D1 V         72         37          1 2015-04-27 15:09:09
REDHAT   EXEC     D1 V         26          9          1 2015-04-17 12:43:45
RESCUE   EXEC     D1 V         26          9          1 2015-04-24 14:53:21
RESCUE   PRM      D1 V         66          7          1 2015-04-24 14:53:05
SLES12   EXEC     D1 V         57         11          1 2015-04-17 12:15:35
SLES12   PARMFILE D1 V         18          2          1 2015-04-25 14:36:02
SWAPGEN  EXEC     D1 V         72        599          7 2013-12-17 21:52:11
SWAPGEN  HELPCMS  D1 V         76        279          3 2013-12-23 10:45:36
SWAPGENH PSBIN    D1 V        256       1632         88 2013-12-23 10:44:21
SWPUME   TEXT     D1 F         80         43          1 2013-12-17 21:52:19
SWPUMEA  TEXT     D1 F         80         47          1 2013-12-17 21:52:19
SWPUMEB  TEXT     D1 F         80         43          1 2013-12-17 21:52:19
SWPUMEC  TEXT     D1 V         80         47          1 2013-12-17 21:52:20
SWPUMED  TEXT     D1 V         80         47          1 2013-12-17 21:52:20
SWPUMEE  TEXT     D1 F         80         57          2 2013-12-17 21:52:20
SWPUMEF  TEXT     D1 V         80         47          1 2013-12-17 21:52:20
SWPUMEG  TEXT     D1 V         80         47          1 2013-12-17 21:52:20
SWPUMEJ  TEXT     D1 V         80         47          1 2013-12-17 21:52:20
```

You can also perform this task by using a special invocation of FILELIST that is used for working with SFS called DIRLIST:

```
===> DIRLIST D
```

5. Create ALIASes for the Linux virtual machines. Each Linux virtual machine will see what appears to be an individual copy of PROFILE EXEC in their SFS directory that is accessed as file mode A. In reality, it will be a read-only pointer back to `LNX:LNXADMIN.PROFILE EXEC`:

```
===> access lnx:linux1. G (forcerw
===> create alias profile exec D profile exec G
===> release G
```

An abbreviated version is used for LINUX2:

```
===> access linux2. G (forcerw
===> create alias profile exec D profile exec G # release G
```

A further abbreviated version is used for LINUX3 and LINUX4:

```
===> acc linux3. G (forcerw # cre ali profile exec D profile exec G # rel G
===> acc linux4. G (forcerw # cre ali profile exec D profile exec G # rel G
```

> **Note:** SFS aliases function in a virtually identical way to symbolic links in UNIX or Linux. Check aliases by using the CMS command **QUERY ALIAS** or by pressing **PF10** while in FILELIST.

The necessary tasks that are under the LNXMAINT ID are complete. Log off the ID to release the remaining accessed directories.

6.17 Creating identity LNXADMIN for Linux administration

Now, create the first *identity* or multi-configuration virtual machine (MCVM), LNXADMIN. An MCVM can be logged on to all members of the SSI at the same time. Therefore, it is not possible to migrate an MCVM between SSI members.

The LNXADMIN virtual machine has many administrative purposes:

- The Linux installation server
- The clone server for cloning from the golden image to target virtual machines
- The Red Hat kickstart server for hosting the necessary files for automated installations
- The administration server for other systems management tools, such as xCAT

To create **LNXADMIN**, perform the following steps while you are logged on as **MAINT** or **MAINT720**:

1. You used the existing profile TCPCMSU when you defined the LNXMAINT user. Now, you create **LNXADMIN** and use the **LNXPDFLT** profile, which is the default user directory profile for Linux virtual machines:

   ```
   ==> xedit LNXADMIN DIRECT A
   ====> input

   IDENTITY LNXADMIN LNX4VM 768M 2G BDEG
      INCLUDE LNXPDFLT
      OPTION LNKNOPAS

   ====> file
   ```

2. Send the entry to DirMaint for processing:

   ```
   ===> dirmaint add lnxadmin
   DVHREQ2289I Your ADD request for LNXADMIN at * has completed; with RC = 0.
   ```

3. Create the sub-configuration entries by using the SUBCON prototypes that were created earlier in 6.13.2, “Role-based access controls and CP privilege classes” on page 206:

   ```
   ===> dirmaint add LNXADM-1 like SUBPRO-1 build on RDBKZVMG in LNXADMIN
   DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT ...
   Ready;
    DVHREQ2288I Your ADD request for LNXADM-1 at * has been accepted.
   ...
   DVHREQ2289I Your ADD request for LNXADM-1 at * has completed; with RC = 0.

   ===> dirmaint add LNXADM-2 like SUBPRO-2 build on RDBKZVMH in LNXADMIN
   ```

```
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT ...
Ready;
 DVHREQ2288I Your ADD request for LNXADM-2 at * has been accepted.
...
DVHREQ2289I Your ADD request for LNXADM-2 at * has completed; with RC = 0.
```

4. Provision two full-pack (the entire usable area of the DASD) minidisks to LNXADMIN for Linux installation. You determined the volumes that you will use on your planning worksheet.

   Remember the following important points:

   – Because this identity is a multi-configuration virtual machine (MCVM) and not a single configuration virtual machine (SCVM), you must assign minidisks to the SUBCONFIG IDs (LNXADM-#) and *NOT* the base ID.

   – If you want to use HYPERPAV, you must assign full-pack minidisks.

   – All Linux and Linux virtual machines use the LNX file pool for their A disk, so you do not need to provision a 191 minidisk.

   – In the example environment that was used to author this book, both Red Hat and SUSE distributions were installed, which facilitated the requirement for the allocation of two full-pack minidisks on each of the two nodes in the SSI cluster. You might not need two full disks on each node. If you are not sure, add them anyway because it is simple to remove them later by using the DirMaint DMDISK command:

   ```
   ===> dirmaint for LNXADM-1 amdisk
   ```

5. Complete the DirMaint AMDISK panel as shown in Figure 6-25.

```
--------------------------------DirMaint AMDISK------------------------------
 To add a new minidisk to a user definition, fill in the following:
   Minidisk Address ===> 0100     Device Type ===> 3390
 Fill in one of the following rows:
   Explicit Start ===>          Size ===>             Volser ===>
   AUTOV                        Size ===> 10016       Volser ===> VM1567
   VBLK   Blksize ===>        Blocks ===>             Volser ===>
   AUTOG                        Size ===>            Grpname ===>
   GBLK   Blksize ===>        Blocks ===>            Grpname ===>
   AUTOR                        Size ===>             Region ===>
   RBLK   Blksize ===>        Blocks ===>             Region ===>
   T-DISK                       Size ===>
   TBLK   Blksize ===>        Blocks ===>
   V-DISK                       Size ===>
   VDBS   Blksize ===>        Blocks ===>
   DEVNO           Real Device Number ===>
 Optionally fill in:
   Link Mode ===> MR
   BLKSIZE   ===>          LABEL ===>
   PWS  Read ===> LNX4VM    Write ===> LNX4VM    Multi ===> LNX4VM    (passwords)
```

Figure 6-25 DirMaint Add MiniDisk Panel for LNXADM-1

   After you complete the fields as shown, press **PF5** to submit.

   Because these minidisks will be Linux minidisks, CMS does not need to format them. By leaving the LABEL field blank, DirMaint does not use CMS to format the minidisks:

```
DVHXMT1191I Your AMDISK request has been sent for processing to DIRMAINT...
DVHSHN3430I AMDISK operation for LNXADM-1 address 0100 has finished
```

Alternatively, you can issue the following command instead of using the DirMaint AMDISK panel:

===> **dirmaint for LNXADM-1 AMDISK 0100 3390 AUTOV 10016 *VM1567* MR PW LNX4VM LNX4VM LNX4VM**

Regardless of whether you use the panel or the line command, the message DVHSHN3430I indicates that the request completed successfully.

6. Assign the 0200 minidisk to LNXADM-1. If you are using the DirMaint line commands, enter these commands:

   ==> **dirmaint for LNXADM-1 AMDISK 0200 3390 AUTOV 10016 *VM1568* MR PW LNX4VM LNX4VM LNX4VM**

7. Assign the 0100 minidisk to LNXADM-2:

   ===> **dirmaint for LNXADM-1 AMDISK 0100 3390 AUTOV 10016 *VM1569* MR PW LNX4VM LNX4VM LNX4VM**

8. Assign the 0200 minidisk to LNXADM-2:

   ===> **dirmaint for LNXADM-2 AMDISK 0200 3390 AUTOV 10016 *VM156A* MR PW LNX4VM LNX4VM LNX4VM**

6.18 Monitoring SFS file pool usage

As any ID that you configured to be an administrator for the LNX file pool, you can review the current usage at any time by using the following commands:

```
===> vmlink maint 193
===> who lnx
                        STORAGE GROUP REPORT FOR LNX:
                                DATE: 04/17/15
                           TIME GENERATED: 15:17:15

                           USERS IN STORAGE GROUP 2

   User        Storage Group  4K Block Limit 4K Blocks Committed Threshold
   ------------------------------------------------------------------------
   LINUX1            2                  500              1-00%       90%
   LINUX2            2                  500              0-00%       90%
   LINUX3            2                  500              0-00%       90%
   LINUX4            2                  500              0-00%       90%
   LNXADMIN          2                30000          16336-54%       90%
   LNXMAINT          2                 5000              0-00%       90%
   MAINT             2                 2000              1-00%       90%


===> tally lnx
                            STATUS REPORT FOR LNX:
                                DATE: 04/17/15
                           TIME GENERATED: 15:26:14

   FILE POOL INFORMATION
           500  MAXIMUM NUMBER OF STORAGE GROUPS
           500  MAXIMUM NUMBER OF MINIDISKS
       6447104  POTENTIAL ADDRESSABLE 4K BLOCKS IN FILE POOL
```

```
CURRENTLY DEFINED MINIDISK INFORMATION
 MINIDISK NUMBER   GROUP NUMBER  4K BLOCKS IN-USE   4K BLOCKS FREE
          1              1              78 -   3%        2611
          2              2           16338 -  18%       73560

PHYSICAL/ALLOCATED BLOCK INFORMATION
 GROUP NUMBER  # of USERS  PHYS. 4K BLOCKS  ALLOC. 4K BLOCKS  DIFFERENCE
      1             0            2689             0                2689
      2             7            89898            39000            50898
```

The installation and configuration of z/VM are complete.

7

z/VM live guest relocation

z/VM 6.2 and later can relocate Linux guests between members in a single system image (SSI) cluster. This capability is known as *live guest relocation* (LGR).

While Linux systems continue to run, they can be moved across logical partitions (LPARs) on the same central processor complex (CPC), or cross-CPC, if the SSI is set up that way. This new function allows for few or even no planned outages.

In this chapter, we provide a brief overview of LGR and information about how to relocate a Linux guest.

This chapter includes the following topics:

7.1 LGR considerations

An SSI cluster features the following types of virtual machines:

- Single-configuration virtual machine

  A virtual machine that is defined by the `USER` statement 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.

- Multi-configuration virtual machine

  A virtual machine that is defined by the `IDENTITY` and `SUBCONFIG` statements can be logged on concurrently to multiple members of the SSI cluster. The virtual machines have common attributes but they can also be configured to access different resources. *Multi-configuration virtual machines are not eligible for guest relocation*.

There are many considerations for relocating running Linux systems.

7.1.1 General considerations before relocation

When you determine the size of a guest that is being relocated, consider the following factors:

- The private virtual disks that the virtual machine can have.
- The potential size to which the guest might grow, including standby and reserved memory (storage) settings.
- The level of memory overcommitment that is on the destination system. Relocation might increase paging demands. Therefore, ensure that at least two times more paging space is available than the total virtual memory across all guests.
- A guideline is to never allow paging space for z/VM to go above 50% full. This rule gives the control program (CP) space to react to sudden increases in central memory demand. Check on this value with the **`CP QUERY ALLOC PAGE`** command. If you add the size of the virtual machine that is being relocated to the pages in use, and that total brings the "in use" percentage over 50%, the relocation might negatively affect system performance.
- Use the **`VMRELOCATE TEST`** command before **`VMRELOCATE MOVE`**.
- The **`SET RESERVED`** setting for the guest (if any) on the source system is not carried over to the destination system. This setting for the guest on the destination must be established after the relocation completes, which is based on the available resources and workload on the destination system.

7.1.2 Mandatory memory checking that is performed during relocation

As part of eligibility checking and in-between memory move passes, relocation ensures that the current memory size of Linux fits in the available space on the destination system:

- For purposes of the calculation, relocation assumes that the Linux memory is fully populated (including the guest's private virtual disks), and includes an estimate of the size of the supporting CP structures.
- Available space includes the sum of available central, expanded, and auxiliary memory.

This check cannot be bypassed. If it fails, the relocation is terminated. The error message that is displayed indicates the size of the guest with the available capacity on the destination system.

7.1.3 Optional memory checking that is performed during relocation

In addition to the mandatory test described, by default, the following three checks are also performed during eligibility checking and in-between memory passes:

- Will the guest's current memory size (including CP supporting structures) exceed auxiliary paging capacity on the destination?
- Will the guest's maximum memory size (including CP supporting structures) exceed the available space (main storage, expanded storage, and auxiliary storage) on the destination?
- Will the guest's maximum memory size (including CP supporting structures) exceed auxiliary paging capacity on the destination?

> **Note:** The maximum memory size includes any standby and reserved memory that the guest might have.

If any of these tests fail, the relocation is terminated. The error message that is displayed indicates the size of the guest with the available capacity on the destination system.

If you are certain that these three checks do not apply to your installation (for instance, because you have an overabundance of central memory and a less than recommended amount of paging space), you can choose for CP to skip these three checks by specifying **`FORCE STORAGE`** on the **`VMRELOCATE`** command.

7.1.4 Minimizing link and resource contention

The relocation process monitors system resources and might determine that a relocation needs to be slowed down temporarily to avoid exhausting system resources. Link and resource contention might negatively affect performance and therefore increase quiesce time during relocation. Therefore, it is recommended that only one relocation is performed at a time. If a set of relocations is to be initiated from a single script or EXEC, you can use the **`SYNC`** option (the default) on the **`VMRELOCATE`** command.

7.2 Relocate a Linux system

You can use the **`VMRELOCATE`** command to move a Linux system from the SSI member on which it is running to another member in the cluster. To accomplish this task, perform the following steps:

1. Log on as `MAINT` on the member where the Linux system is running. In this example, the Linux system `LINUX1` is running on member 1, `ITSOZVM1`.
2. Choose a sample Linux system to relocate and verify that it is running on the member. In this example, the target is `LINUX1`:

   ```
   ===> q LINUX1
   LINUX1 - DSC
   ```

   The output shows that `LINUX1` as disconnected, which means that it is running on *this member*.

3. Issue the **VMRELOCATE TEST** command with a target of the second SSI member to test whether the system is eligible for relocation:

```
===> vmrelo test linux1 ITSOZVM2
User LINUX1 is eligible for relocation to ITSOZVM2
Ready; T=0.01/0.01 10:52:06
```

4. You might choose to start a **ping** from another session. For example, to **ping** continuously from a DOS session, issue the following command:

```
c:\>ping /t vmlnx2-1.itso.ibm.com

Pinging virtcook1.itso.ibm.com [9.12.7.1] with 32 bytes of data:

Reply from 9.12.7.96: bytes=32 time=4ms TTL=64
Reply from 9.12.7.96: bytes=32 time=3ms TTL=64
Reply from 9.12.7.96: bytes=32 time=3ms TTL=64
...
```

5. Issue the **VMRELOCATE MOVE** command to migrate the running Linux system:

```
===> vmrelo move linux1 itsozvm2
Relocation of LINUX1 from ITSOZVM1 to ITSOZVM2 started
User LINUX1 has been relocated from ITSOZVM1 to ITSOZVM2
```

6. Monitor the **ping** session to see whether packets are delayed or dropped.
7. Verify that the Linux system is now running somewhere in the SSI:

```
===> q LINUX1
LINUX1 - SSI
```

   The output shows `LINUX1` as `SSI`, which means that it is running on *a different member*.

We described how to migrate a running Linux system by using the **VMRELOCATE** command.

8

Servicing z/VM

This chapter focuses on the requirements to keep your z/VM systems updated to ensure full functionality, optimal utility, security, and the elimination of known problems. The process of ordering and applying z/VM Service also is described.

The full details are provided to apply the two main types of service:

- Recommended service upgrade (RSU), which is analogous to a service pack
- Program temporary fix (PTF), which is analogous to a bug fix

The processes to install these types of service are essentially the same.

> **Awareness is key:** Go to the z/VM website and subscribe to notifications for Service News and Red Alerts. For more information, see 6.5.5, “Service-level validation and subscribing to service notifications” on page 154.

This chapter includes the following topics:

8.1 z/VM release schedule

Beginning with z/VM 7.1, a new z/VM release will be delivered on a fixed, 24-month cycle. Each release contains all previously released New Function APARs, which are functions that are too disruptive to ship in the service stream, and all fixes shipped on the service stream for the previous release.

A z/VM release can be ordered for 18 months after the general availability of its follow-on release. IBM will provide service to a z/VM release until six months after the general availability of its N+2 release.

This release schedule allows customers to update to the current release and access new functions, or keep the current version and receive only the corrective fixes. For example, z/VM 6.4 users receive corrective service six months after the general availability of z/VM 7.2. This schedule gives each z/VM release a life span of 54 months when IBM provides service and support.

To track the current and planned enhancements for the new release cycle, IBM created a web page on the z/VM website to list them. At this web page, you can subscribe to receive updates about new functions or to specific APARs (see Figure 8-1).

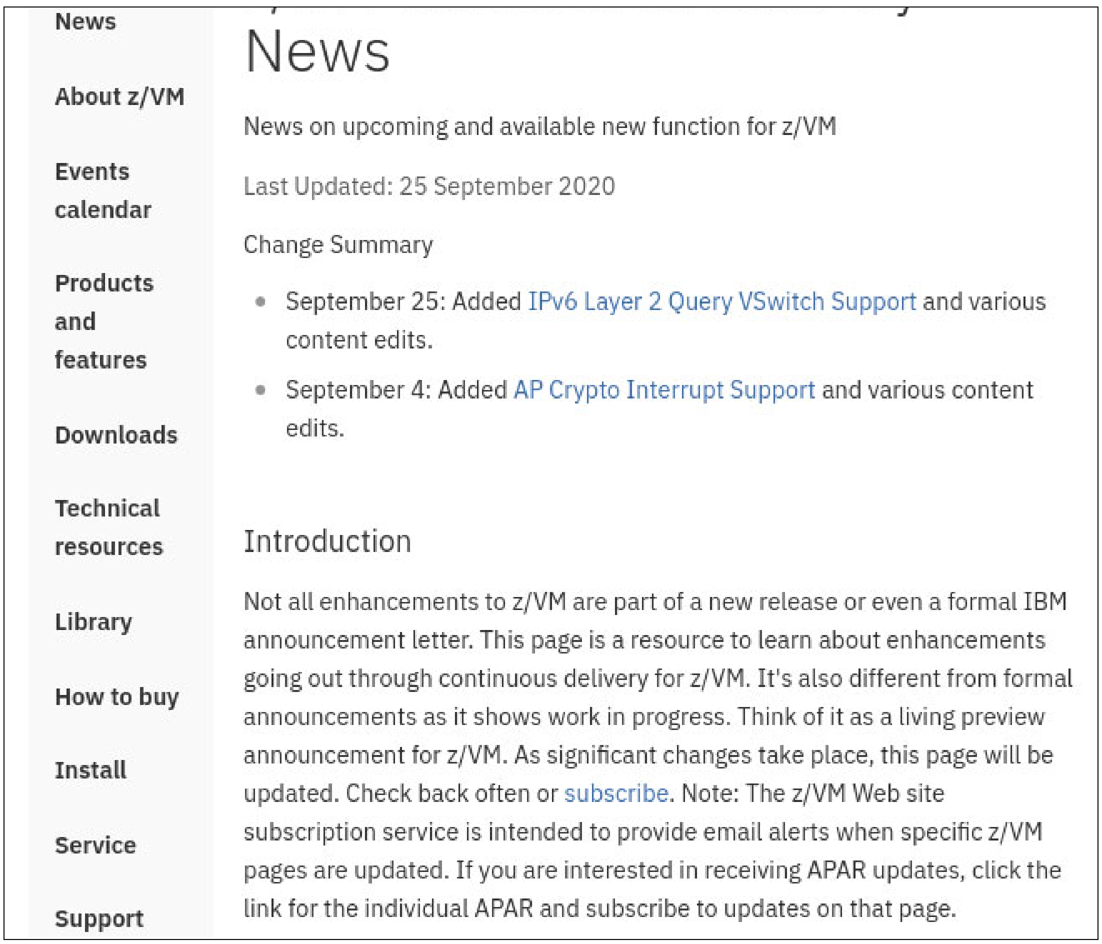


Figure 8-1 z/VM Continuous Delivery News

8.2 Recommended service upgrades

IBM provides recommended maintenance service for all components, products, and features that are delivered with the z/VM base system in a single package that is called a *recommended service upgrade* (RSU). An RSU contains cumulative service in a pre-built format. Customers are advised to maintain RSU currency of a minimum of six months on their production z/VM systems. Customers must install only the latest available RSU to keep currency.

RSUs ("stacked" or otherwise) are packages, named *vrnn* - version, release, and a sequence number. For example, RSU 6204 is the fourth RSU for z/VM 6.2. You can get the latest RSU for a release by ordering special program temporary fix (PTF) number UM97vr0, where *vr* is the version and release. Inside the RSU is a collection of one or more service levels.

A *service level* (SL) is a tested subset of all of the available PTFs and is named *yynn*, where *yy* is the year of issue and *nn* is a sequence number. This sequence number has nothing to do with the RSU sequence number; therefore, they likely do not match. Within each release, a single SL is established for the following parts of z/VM:

- The base: Control program (CP), Conversational Monitor System (CMS), and so on
- TCP/IP
- RACF
- PerfKit
- DIRMAINT
- Remote Spooling Communications Subsystem (RSCS)
- Hardware Configuration Definition (HCD)

When it is time to deliver a new RSU, the RSU sequence number is incremented and all of the available service levels for that release are placed in it. At least one of them is new, but the others are the same as on the previous RSU. Service levels are cumulative. They contain all of the PTFs that were in the earlier service levels for that release.

> **Important:** When you apply service, you might want to back it up. It is recommended that an up-to-date backup of your system exists before you start the next tasks.

For more information about applying corrective service to z/VM, see the following manuals:

- *z/VM Guide for Automated Installation and Service* (see Part 4), GC24-6197
- *z/VM Service Guide*, GC24-6232

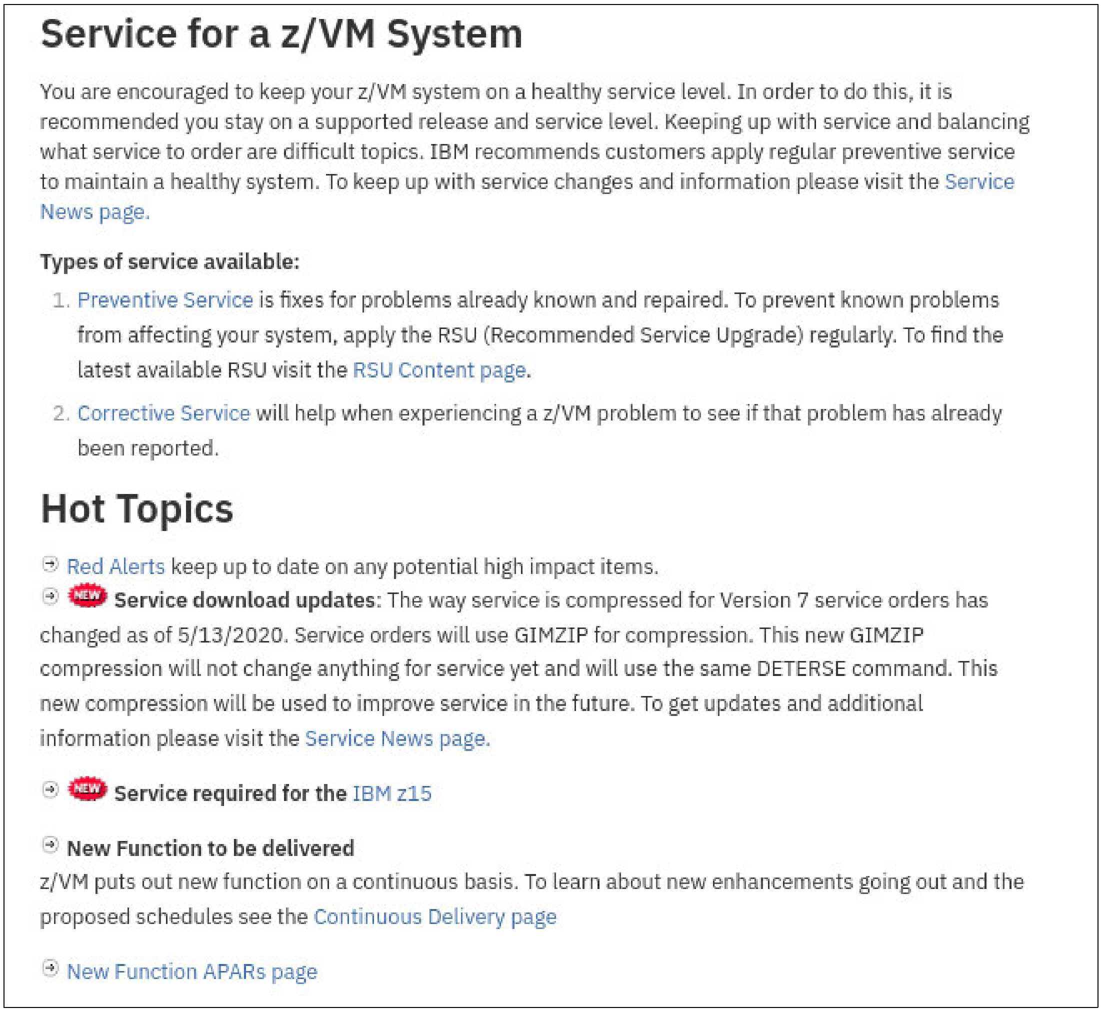
These manuals are more complete than this chapter. You might consider the use of these manuals first, rather than this chapter. At least, use them as references.

VMSES/E is a component of z/VM that provides the **SERVICE** and **PUT2PROD** EXECs. The **SERVICE** EXEC performs the following functions:

- Installs an RSU or applies corrective (COR) service for z/VM components, features, or products.
- Displays either the RSU level of the specified component or whether a particular PTF or authorized program analysis report (APAR) was applied (when used with STATUS).
- Creates PTF bitmap files (when used with BITMAP) that contain a list of all PTFs that were received, applied, or superseded, and all installed products.

When **SERVICE** is successfully completed, the **PUT2PROD** EXEC places the z/VM components, features, or products that are installed on the z/VM system deliverable, and serviced, into production. For more information, see this web page.

The body of the page is similar to the example that is shown in Figure 8-2.



Service for a z/VM System

You are encouraged to keep your z/VM system on a healthy service level. In order to do this, it is recommended you stay on a supported release and service level. Keeping up with service and balancing what service to order are difficult topics. IBM recommends customers apply regular preventive service to maintain a healthy system. To keep up with service changes and information please visit the Service News page.

Types of service available:

1. Preventive Service is fixes for problems already known and repaired. To prevent known problems from affecting your system, apply the RSU (Recommended Service Upgrade) regularly. To find the latest available RSU visit the RSU Content page.
2. Corrective Service will help when experiencing a z/VM problem to see if that problem has already been reported.

Hot Topics

Red Alerts keep up to date on any potential high impact items.

NEW **Service download updates**: The way service is compressed for Version 7 service orders has changed as of 5/13/2020. Service orders will use GIMZIP for compression. This new GIMZIP compression will not change anything for service yet and will use the same DETERSE command. This new compression will be used to improve service in the future. To get updates and additional information please visit the Service News page.

NEW **Service required for the** IBM z15

New Function to be delivered

z/VM puts out new function on a continuous basis. To learn about new enhancements going out and the proposed schedules see the Continuous Delivery page

New Function APARs page

Figure 8-2 z/VM Service main web page

Consider visiting several of the links on this page.

8.3 Applying a recommended service upgrade

Applying an RSU is similar to applying a PTF, which was described in the previous section. z/VM service can be preventive (RSU) or corrective (COR).

For more information about the latest RSU content, see this web page.

For more information about Red Alerts, which contain information about potential high-impact items, see this web page.

Next, we summarize applying service and also describe how to obtain service over the internet by using IBM Shopz.

First, you must determine whether your system needs service by using the **QUERY CPLEVEL** command:

```
==> query cplevel
z/VM Version 7 Release 1.0, service level 1801 (64-bit)
Generated at 09/10/20 17:34:33 EDT
IPL at 09/11/20 16:34:30 EDT
```

The *service level* (also called *RSU level*) is a four-digit field that consists of a pair of two-digit segments. The first two digits represent the calendar year that the RSU was published, and the second two digits represent the sequential RSU level within that year. Examples are `1902RSU`, which is the second RSU for calendar year 2019, and `2001RSU`, which is the first RSU of calendar year 2020.

Use the following overall steps in applying an RSU:

1. Getting service from the internet.
2. Downloading the service files.
3. Receive, apply, and build the service.
4. Putting the service into production.

8.3.1 Getting service from the internet

An RSU is obtained by its PTF number. The PTF for the most current RSU is of the form **UM97*xyz*** where *xyz* is the z/VM version-release-modification level. So for z/VM 6.3, the RSU is `UM97630`, and for z/VM 7.2, it is `UM97720`.

With Shopz, you do not need to know the PTF number. If you know that you want the latest RSU, you can get it directly, based on the version of z/VM that you are running.

Complete the following steps, which are the same steps that are described with some figures in 8.4, “Applying a program temporary fix” on page 265:

1. Point a web browser to the Shopz website.
2. Click **Sign In/Register**, which often is in the upper left. Use your user ID and password, if you have them. If not, click **Create IBMid** and complete the form to create a user ID and password. You must have your IBM customer number.
3. Click **My orders** near the top.
4. The My Orders page opens. Click **Create orders**, select **z/VM - Service**. Choose **RSU Recommended Service Upgrade** in the drop-down menu. Click **Continue**.

5. The next windows of forms are self-explanatory. In the window asking for the running version, choose the radio button that applies to your version of z/VM. In this example, we used z/VM Version 7.1.0 Stacked 7104RSU (PTF UM97710).
6. On the window for specifying the delivery mechanism, choose **Internet**.
7. On the next window, verify whether all order information is correct and click **Submit**.
8. In a few minutes, you receive two emails: one is for the core RSU, the other is for the *PSP bucket* (more fixes after the RSU). Alternatively, you can click the refresh button on your browser. After a brief period, the Status changes to a link named Download, as shown in Figure 8-3. Click **Download**.

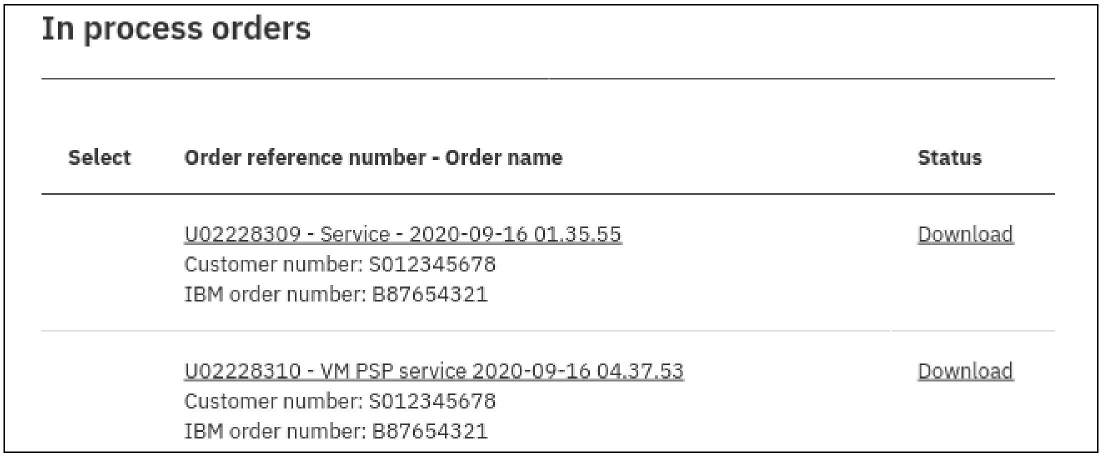


Figure 8-3 Downloading service directly from your browser

8.3.2 Downloading the service files

In this example, the service files are staged on a desktop machine, then copied to z/VM with FTP. We assume you configured the `LOGONBY` facility as recommended for increased security and accountability; therefore, you log in as `ADM123` by using its own password.

`MAINT720` features a special disk that is used for storing service packages, the 500 disk. This disk have 1200 cylinders by default, and should have enough space to store both the compressed and decompressed service files.

Complete the following steps:

1. Download the files to your desktop or another staging system. This example has two files. The `SHIPTFSS` file is for the PSP bucket, and the `SHIPRSU1` file is for the RSU.
2. Use FTP to send the file to the `MAINT720 500` disk. The following example shows a Windows command-line FTP session:

```
C:\Downloads>ftp 9.10.11.12
User (9.10.11.12:(none)): maint720.by.adm123
Password:
ftp> cd maint720.500
250 Working directory is MAINT720 500
ftp> bin
200 Representation type is IMAGE.
ftp> quote site fix 1024
200 Site command was accepted.
ftp> put S9338801.shiptfss
...
```

```
ftp> put S9338766.shiprsu1
...
ftp> quit
```

3. Log on to MAINT720.
4. Access the MAINT720 500 disk as file mode C. Query the disks:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
==> query disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
MNT191 191  A   R/W   175 3390 4096       26        231-01      31269
31500
MNT5E6 5E6  B   R/W     9 3390 4096      131       1265-78        355
1620
MNT500 500  C   R/W   900 3390 4096        2      50705-31     111295
162000
MNT51D 51D  D   R/W    26 3390 4096      299       1731-37       2949
4680
PMT551 551  E   R/W    40 3390 4096        9         92-01       7108
7200
MNT190 190  S   R/O   207 3390 4096      694      16694-45      20566
37260
MNT19E 19E  Y/S R/O   500 3390 4096     1126      29765-33      60235
90000
```

5. List the files on the C disk and note the two new files:

```
==> listfile * * c
S1309082 SHIPRSU1 C1
7201RSU1 SERVLINK C1
S1309082 SHIPDOC  C1
```

6. De-terse the documentation file. Change the file name prefix character to “d”:

```
==> deterse s1309082 shipdoc c d1309082 = =
```

7. De-terse the RSU file. Change the file type to SERVLINK (this step can take some time to complete):

```
==> deterse s1309082 shiprsu1 c = servlink =
```

   Often, this step succeeds. However, large RSUs can fill up the MAINT 500 disk on the **FTP** step or the **DETERSE** step. For example, you might receive the following error on the **DETERSE** step:

```
DMSERD107S Disk C(500) is full
 No traceback - not enough CTL storage
```

   If this error occurs, an extra step to create a larger disk might be necessary.

8.3.3 Receive, apply, and build the service

You must receive, apply, and build the service. Then, it can be put into production.

In the past, this process was lengthy and detailed. For example, to receive, apply, and build the CP component, the following steps were needed:

```
vmfmrdsk zvm cp apply (setup
vmfsetup zvm cp
vmfpsu zvm cp
vmfins install ppf zvm cp (nomemo env {filename} nolink override no
vmfapply ppf zvm cp (setup
vmfbld ppf zvm cp (status
vmfbld ppf zvm cp (serviced
```

Then, the same steps were needed for many other components. The process is easier now with the **SERVICE ALL** command. Alternatively, the previous method is more granular and better enables a system administrator to know the parts of the service that were applied.

Complete the following steps:

1. Log on to a 3270 session as MAINT720.
2. Access the MAINT720 500 disk as C:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
```

3. Apply the service with the **SERVICE ALL** command. The RSU must be applied first (**S8873950 SERVLINK** in this example). Then, apply any PTFs that came after the RSU:

```
==> service all s1309082
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed with warnings
Ready(00004); T=*.**/*.** **:**:**
```

A return code of 0 is ideal. If the last Ready line has a number in parentheses, that number is the return code. In general, a return code of 4 is acceptable, which means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered. View details with the **VMFVIEW SERVICE** command:

```
==> vmfview service
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 7 <===> Number of messages not shown = 764
**********************************************************************
****            SERVICE              USERID: MAINT720            ****
**********************************************************************
****           Date: 09/17/20           Time: 15:34:38           ****
**********************************************************************
CK:VMFSUI2104I PTF UM33449 contains user information. Review the :UMEMO
CK:            section in file UM33449 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (I) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTM MSIBIN
CK:VMFSRV1233I The following products have been serviced.
CK:VMFSRV1233I CMS CP TCPIP VMHCD
```

For these example warnings, if you are running HCD, as the `VMFBDC2250W` message states, you must copy the stated objects to your workstation.

4. Press **F3** to exit **`XEDIT`**.
5. IPL CMS again and press **Enter** at the `VM READ` prompt:

```
==> ipl cms
DMSACC724I 19E replaces Y (19E)
DMSACP723I Y (19E) R/O
z/VM V7.2.0    2020-09-17 17:56
...
```

6. Access the `MAINT720 500` disk again as `C`:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
```

7. Apply the PSP bucket, if one exists (in this example, no PSP bucket existed for RSU7201; therefore, an older PSP bucket is shown):

```
==> service all S9338801
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed with warnings
Ready(00004); T=29.96/33.46 15:55:40
```

In this example, the service was installed, but warnings were shown.

8. Run the **`VMFVIEW SERVICE`** command:

```
==> vmfview service
    ===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 1 <===> Number of messages not shown = 510
************************************************************************
****             SERVICE              USERID: MAINT720             ****
************************************************************************
****            Date: 09/17/20          Time: 15:53:09             ****
************************************************************************
RO:VMFAPP2112W PTF UK59536 has a IFREQ requisite for PTF UM33113 in
RO:            product 7VMCMS20 (CMS component for z/VM 6.1.0)
* * * End of File * * *
```

This message states that a relationship exists between the two PTFs (`UM33113` and `UK59536`). It is advisable to ensure that you have both PTFs, or know about the requisite PTF and decide whether it is important in your environment.

9. Press **F3** to exit **`XEDIT`**.
10. Log off from `MAINT720`.

8.3.4 Putting the service into production

In this section, we describe how to use the **`PUT2PROD`** command to put the service into production. Until this point, all the applied fixes are stored on a special set of disks that is called *staging disks*. The **`PUT2PROD`** script transfers those files to the production disks.

Important: The **PUT2PROD** command affects your production environment. It is recommended that all users are logged off before you run it. Placing service into production must be performed as part of a planned system outage because a **SHUTDOWN REIPL** is recommended after you place service into production.

Complete the following steps:

1. Log on to MAINT720 on the first member.
2. IPL CMS:

```
==> ipl cms
z/VM V7.2.0    2020-09-17 17:59
...
```

3. Use the **PUT2PROD** command to put the service into production. Many windows scroll by (this command takes time to complete):

```
==> put2prod
...
VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ the new
            service.
VMFP2P1239I CMS was serviced. Re-IPL CMS in all virtual machines running CMS to
            employ the new service.
VMFP2P2760I PUT2PROD processing completed successfully
```

4. Review the messages by using the **VMFVIEW PUT2PROD** command:

```
==> vmfview put2prod
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 4 <===> Number of messages not shown = 436
************************************************************************
****      PUT2PROD      SYSTEM: LEFT720       USERID: MAINT720      ****
************************************************************************
****            Date: 09/17/20           Time: 18:16:35             ****
************************************************************************
CK:VMFP2P1233I The following products have been put into production.
CK:            Recycle the appropriate servers.
CK:VMFP2P1233I CMS CP TCPIP VMHCD
CK:VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ
CK:            the new service.
CK:VMFP2P1239I CMS was serviced. Re-IPL CMS in all virtual machines
CK:            running CMS to employ the new service.
```

   In this example, the messages are informational. If warning or error messages exist, you must address those issues.

5. Press **F3** to exit **XEDIT**.
6. Although the service was “put into production”, the **QUERY CPLEVEL** command still returns the current service level, which in this example is 2001 (the first RSU in the year 2020) because the new CP load module (nucleus) was not loaded:

```
==> query cplevel
z/VM Version 7 Release 2.0, service level 2001 (64-bit)
Generated at 07/29/20 16:50:40 EDT
IPL at 09/09/20 11:09:59 EDT
```

7. Run the same **PUT2PROD** command on all other members of the single system image (SSI) cluster.

8. To load the new CP load module, shut down and IPL the single system image (SSI) cluster again:
   a. Log off from `MAINT720`.
   b. Log on to `MAINT`.
   c. Issue the **`SHUTDOWN REIPL`** command to restart z/VM and load the new CP nucleus:

      ```
      ==> shutdown reipl
      ...
      ```

   When your system comes back up, it is at the new CP service level.
9. After the system comes back up, start a new 3270 session and log on as `MAINT` on the first member.
10. Run the **`QUERY CPLEVEL`** command again:

    ```
    ==> query cplevel
    z/VM Version 7 Release 2.0, service level 2002 (64-bit)
    Generated at 09/12/20 16:50:40 EDT
    IPL at 09/17/20 11:09:59 EDT
    ```

    This information shows that the new CP load module is used and that the service level is the second RSU in the year 2020.

8.4 Applying a program temporary fix

You might determine that you need to apply a specific fix or *program temporary fix* (PTF) to your system; for example, an APAR, VM6643 was opened when a problem was identified with a cross-system link.

The APAR was assigned the following PTF numbers for each of the following z/VM releases:

z/VM 6.4	UM35514
z/VM 7.1	UM35515
z/VM 7.2	UM35728

Therefore, for z/VM 7.2, you apply PTF UM35728. The following sections provides an example of how to apply this PTF.

8.4.1 Getting service by using Shopz

Service for z/VM is not available on tape media since July 16, 2018 because most of the user base moved to more modern technology. The most used method for getting service is downloading over the internet, which is a more convenient and faster method.

For places where an internet download cannot be used or a physical media is required, IBM still can ship service on DVD.

To download service by using internet delivery, complete the following steps:

1. Point a browser to this web page.
2. Enter the APAR number in the Search For: text field. In this example, the APAR is `UM35728`, and one match was found, as shown in Figure 8-4.

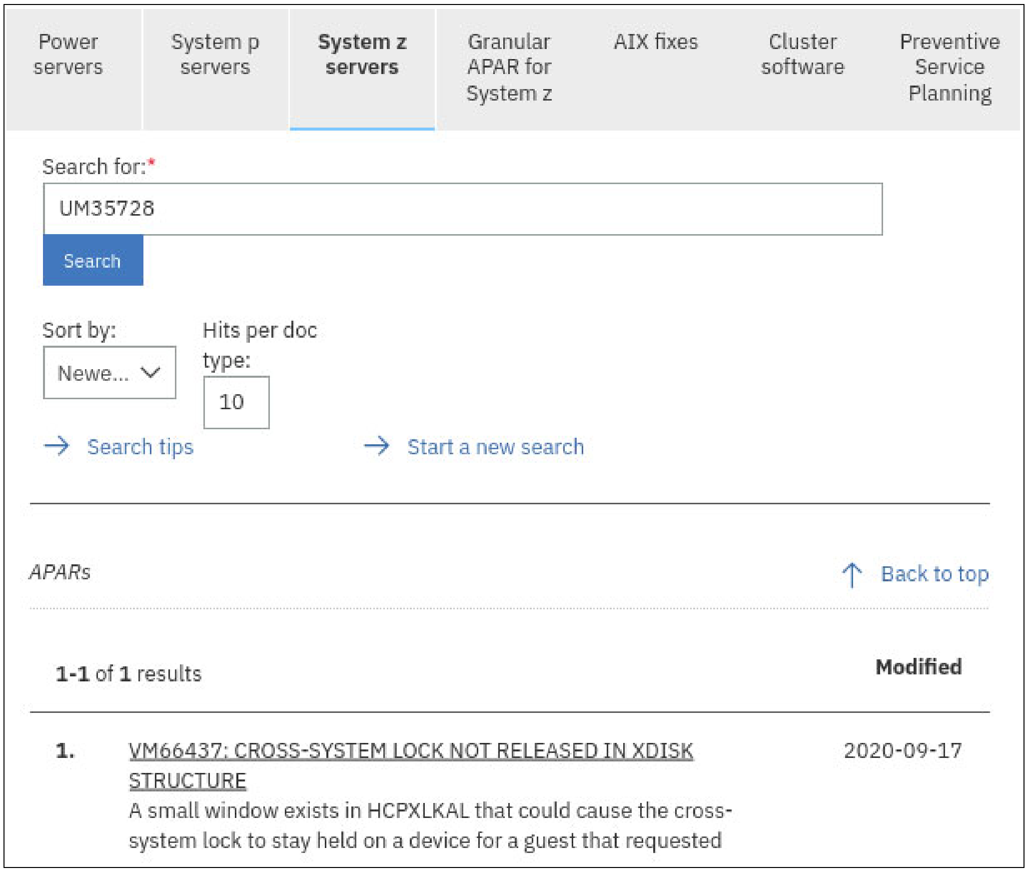


Figure 8-4 Searching for PTFs by APAR number

3. Click the APAR description link.
4. Farther down on the page, note the Fixed component name, which is important. In this example, it is VM CP.

At the bottom of the page, the Applicable component levels section shows that PTF UM35728 is available for z/VM 7.2. Before you get that PTF, ensure that it was not applied by seeing 8.4.2, “Determining whether a PTF was applied” on page 267.

8.4.2 Determining whether a PTF was applied

Check to ensure that the PTF was not previously applied. In this example, we check for the PTF UM33539. Complete the following steps:

1. Log on to `MAINT720`.
2. Use the **SERVICE ALL STATUS** command followed by the PTF number so that you can query whether it was applied:

```
==> service all status UM35728
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE ALL STATUS UM35728
VMFSRV2760I SERVICE processing started
DASD 0491 LINKED R/W; R/O BY    10 USERS
DASD 0492 LINKED R/W; R/O BY    10 USERS
DASD 019D LINKED R/W; R/O BY    17 USERS
DASD 0402 LINKED R/W; R/O BY    13 USERS
DASD 193C LINKED R/W; R/O BY    16 USERS
DASD 0200 LINKED R/W; R/O BY     2 USERS
DASD 0201 LINKED R/W; R/O BY PERSMAPI at ZVM72A
DASD 01CC LINKED R/W; R/O BY PERSMAPI at ZVM72A
DASD 029D LINKED R/W; R/O BY     2 USERS
VMFSRV1227I UM35728 is not received or applied
VMFSRV2760I SERVICE processing completed successfully
```

This message shows that PTF UM35728 was *not* applied. Obtaining and applying this PTF is described next.

8.4.3 Downloading the service to z/VM

You can download a specific APAR (or a list of APARs) by accessing the IBM Software Shopz website and creating an order. The process is similar to ordering a RSU.

On the My Orders page from Shopz, select **Create a new order** (see Figure 8-5).

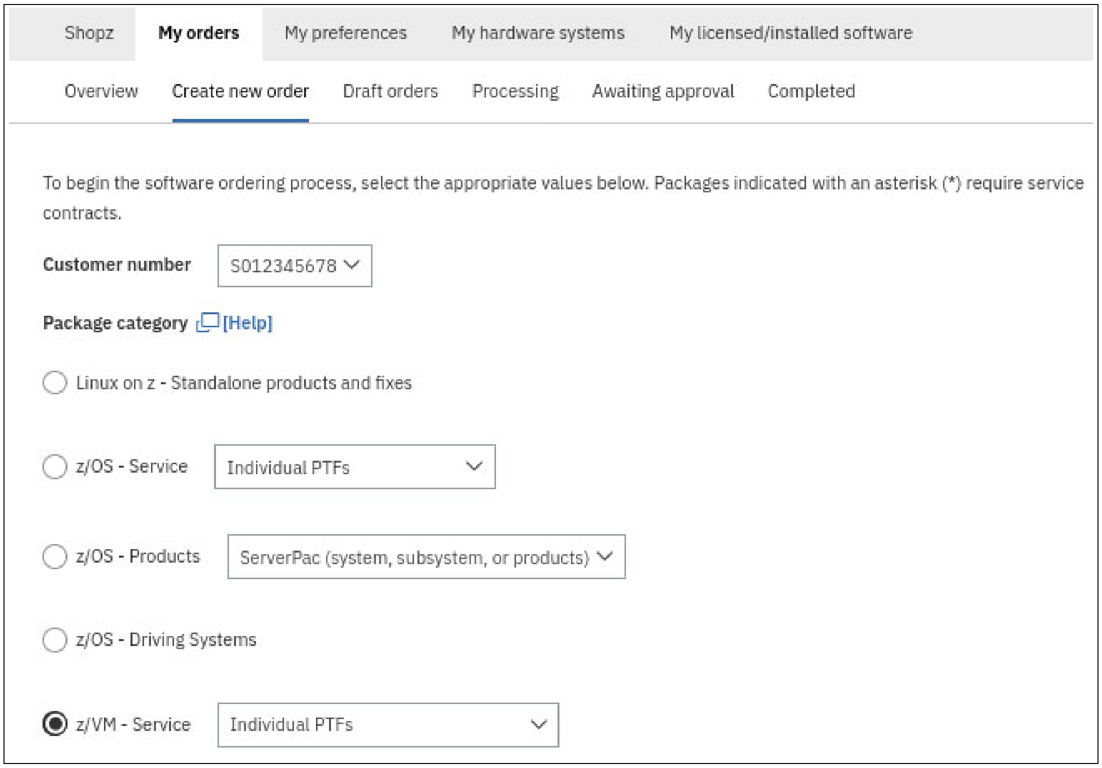


Figure 8-5 Getting specific fixes from Shopz

Complete the following steps:

1. In this example, select **z/VM - Service, Individual PTFs**.
2. In the next window, select the z/VM version and specify whether you want to search by PTF number or APAR number.
3. In the panel, you are prompted for the PTF or APAR list. Enter all of the required PTFs or APARs (separated by blanks or commas). In the next window, select **Internet** as the preferred delivery media. Confirm the order contents, and click **Submit**.
4. You receive an email within a few minutes. The email shows your order number and a link that is used to download the service files. The following example shows the important information in the email:

```
From:    Oms Client01/Boulder/IBM
Subject: IBM Order <Bxxxxxxx> is ready for download.
...
To access your order directly, go to:
https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp?action=downlo
ad&orderId=<Uxxxxxxxd>0
```

5. Point your browser to the link in the email. You see a web page that looks similar to the example that is shown in Figure 8-6.

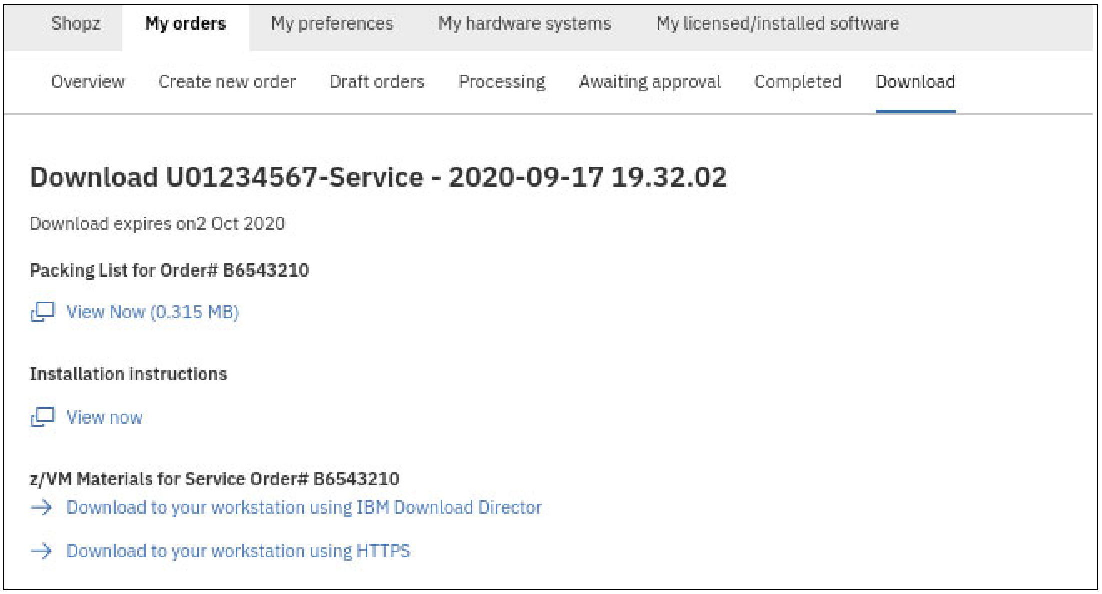


Figure 8-6 Web page that was created for downloading a PTF

6. Choose a method of downloading the files to a desktop or staging machine. In this example, **Download to your workstation using HTTPS** was used. Two small files (a XML and XSL in EBCDIC format, with details of the order) and some large files with `SHIPDOCS.pax.Z` and `SHIPFTSS.pax.Z` extension are available. This new packing format can be extracted by using the **`DETERSE`** command.
7. Send all files to z/VM in binary with fixed 1024-byte records to the `MAINT 500` disk. Usually, FTP is used. While you are downloading the files, note the file sizes. The following example shows a Windows command-line FTP session:

```
C:\downloads> ftp 9.10.11.12
User (9.10.11.12:(none)): maint720.by.adm123
Password:
...
ftp> cd maint720.500
250 Working directory is MAINT720 500
ftp> bin
200 Representation type is IMAGE.
ftp> quote site fix 1024
200 Site command was accepted.
ftp> mput *.pax.Z
150 Storing file 'S00001.SHOPZ.S12345678.SHIPDOCS.pax.Z'
250 Transfer completed successfully.
ftp: 6144 bytes sent in 0.00Seconds 6144000.00Kbytes/sec.
mput S1041690.SHIPTFSS? y
150 Storing file 'S00001.SHOPZ.S12345678.SHIPFTSS.pax.Z'
250 Transfer completed successfully.
ftp: 10240 bytes sent in 0.00Seconds 10240000.00Kbytes/sec.
ftp> quit
```

8. Log on to z/VM as `MAINT720`.

9. Access the MAINT720 500 disk as C:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
```

10. Verify that the files exist by using the **LISTFILE** command:

```
==> listfile * * c
S1041690 SHIPDOCS C1
S1041690 SHIPTFSS C1
7201RSU1 SERVLINK C1
```

11. The envelope files arrive in a compressed format to speed downloads. To use them, you must first rename them to a file type of SERVLINK and decompress them by using the **DETERSE** command. Therefore, it is recommended to leave the file name of the SES envelope unchanged and change the prefix letter of the documentation envelope to D. First, rename them, and then, use the **DETERSE** command with the **(REPLACE** parameter to decompress them in place and save disk space:

```
==> rename s1041690 shiptfss c = servlink =
==> rename s1041690 shipdocs c d1041690 servlink =
==> deterse s1041690 servlink c = = = (replace
==> deterse d1041690 servlink c = = = (replace
```

Ensure that all commands complete successfully.

8.4.4 Receiving, applying, and building the service

You must receive, apply, and build the PTF. Then, it can be put into production. The process is much easier now by using the **SERVICE** command.

To prepare to use the **SERVICE** command, you must have a minidisk with significant free space. Use the MAINT720 500 minidisk for this purpose. Complete the following steps:

1. Access the MAINT720 500 disk as file mode C:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
```

2. Use the **SERVICE ALL** command and specify the envelope files that you downloaded. Many windows of output will scroll by and automatically be cleared. Important messages are saved to the 500 disk. This process can take time. The following example shows the processing:

```
==> service all d1041690
...
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
==> service all s1041690
...
VMFSRV1233I The following products have been serviced.
VMFSRV1233I CP
VMFSRV2760I SERVICE processing completed successfully
```

If you see no number in parentheses after the Ready; prompt, the return code is 0. Any nonzero return code will be in parentheses. A return code of 0 is ideal. In general, a return code of 4 is acceptable. It means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

3. The output files are of the form `$VMF* $MSGLOG`. You might want to inspect these files:

```
==> filel $vmf* $msglog
$VMFSRV  $MSGLOG  A1 V        80       1582         29  1/31/12 15:19:27
      $VMFBLD  $MSGLOG  A1 V        80        841         12  1/31/12 15:19:25
      $VMFAPP  $MSGLOG  A1 V        80        212          3  1/31/12 15:19:15
      $VMFREC  $MSGLOG  A1 V        80         69          1  1/31/12 15:19:15
      $VMFMRD  $MSGLOG  A1 V        80        270          4  1/31/12 15:19:14
      $VMFINS  $MSGLOG  A1 V        80        223          4 11/29/11  2:32:50
      $VMFP2P  $MSGLOG  A1 V        80       1741         32 11/29/11  0:55:22
```

4. Run the **VMFVIEW SERVICE** command to review the results of the previous **SERVICE** command. Press the **F3** key to quit. The following example shows the **VMFVIEW SERVICE** command:

```
==> vmfview service
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 2 <===> Number of messages not shown = 126
**********************************************************************
****             SERVICE              USERID: MAINT720            ****
**********************************************************************
****            Date: 09/17/20          Time: 15:19:13            ****
**********************************************************************
CK:VMFSRV1233I The following products have been serviced.
CK:VMFSRV1233I CP
```

Ideally, the process produces no output. If errors occur, you must address them. If warnings occur, they might be acceptable, but you still must investigate them.

8.4.5 Putting the service into production

To put the service into production, complete the following steps:

1. Log on as `MAINT720`.
2. IPL CMS:

```
==> ipl cms
z/VM V7.2.0    2020-09-17 15:26
```

3. Access the VMSES/E test build disk as file mode `B`:

```
==> access 5e6 b
DMSACC724I 5E6 replaces B (5E6)
```

4. Use the **PUT2PROD** command to put the service into production:

```
==> put2prod
...
VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ the new
            service.
VMFP2P2760I PUT2PROD processing completed successfully
```

The second to last message states that a **SHUTDOWN** and re-IPL are necessary. Again, watch for a return code of `0`.

5. Your PTF is now *put into production.* You might or might not need to IPL the system, depending on the nature of the PTF that you applied. If necessary, ensure that you can IPL your system again. You might want to shut down and IPL again one member at a time with live guest relocations (LGRs) of the important Linux systems in between.

6. Your z/VM system returns in a few minutes. When the system comes back up, start a 3270 session to MAINT and again query the status of the PTF:

```
==> service cp status UM35728
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE CP STATUS UM35728
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (7VMCPR20%CP) PTF UM35728 status:
VMFSRV1226I    RECEIVED  09/17/20 15:19:15
VMFSRV1226I    APPLIED   09/17/20 15:19:15
VMFSRV1226I    BUILT     09/17/20 15:19:27
VMFSRV1226I    PUT2PROD  09/17/20 15:24:46 POKDEV72
VMFSRV2760I SERVICE processing completed successfully
```

   This query shows that the PTF was successfully applied.

7. Repeat the steps in this section for all members in the SSI cluster.

8.4.6 Checking for APARMEMO files

After you apply PTFs, check for files with a file type of APARMEMO on the MAINT720 500 disk. These files might provide instructions for more work after the PTFs are applied.

Complete the following steps:

1. Access the MAINT 500 disk as C and list the files with file type APARMEMO:

```
==> access 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * aparmemo c
7VMCPR20 APARMEMO C1
```

   This example shows one APARMEMO file.

2. Review the contents of the file:

```
==> type 7vmcpr20 aparmemo c

APAR MEMOS      09/17/20.16:16:55
=================================

        THE FOLLOWING MEMOS WERE INCLUDED WITH THE PTFS SHIPPED:

        NONE.
```

In this example, the APARMEMO file was created, but no extra memos are present.

You do not see any new information in the APARMEMO file if you did not perform **SERVICE** against the documentation SERVLINK file because the <*prodid*> MEMO file is in the documentation SERVLINK file.

8.5 Determining the TCP/IP service level

Often, you need to query more than the service level. We took the following steps from the CP Maintenance Levels and Virtual Switch TCP/IP Maintenance Levels links, starting at this web page.

Complete the following steps:

1. Log on to `TCPMAINT` on one of the SSI members. Use the **`QUERY VMLAN`** command to determine the latest APAR that was applied:

```
==> query vmlan
VMLAN maintenance level:
  Latest Service: Base
VMLAN MAC address assignment:
  System MAC Protection: OFF
  MACADDR Prefix: 020211 USER Prefix: 020210
  MACIDRANGE SYSTEM: 000001-FFFFFF
             USER:   000000-000000
VMLAN default accounting status:
  SYSTEM Accounting: OFF      USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 4
  TRANSIENT  Limit: INFINITE   Current: 0
Trace Pages: 8
VMLAN Directory Network Authorization: ENABLED
IVL Domain: None
```

   The `Latest Service: Base` line shows that no APAR was applied. The maintenance level of the TCP/IP stack is important to virtual networking.

   To determine this maintenance level, first get the active virtual switch controller:

```
==> query vswitch vswitch1
VSWITCH SYSTEM VSWITCH1   Type: QDIO    Connected: 1    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED LOCAL
  VLAN Unaware
  MAC address: 02-02-11-00-00-01    MAC Protection: Unspecified
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Ready                         PriQueuing: OFF
  PMTUD setting: EXTERNAL   PMTUD value: 9000     Trace Pages: 8
  Portname: OSA1948  RDEV: 1940.P01 Controller: DTCVSW1  VDEV: 0600 ACTIVE
        Adapter ID: 3907000BB4B7.0148
  Portname: OSA1958  RDEV: 1950.P01 Controller: DTCVSW3  VDEV: 0600 BACKUP
        Adapter ID: 3907000BB4B7.0130
```

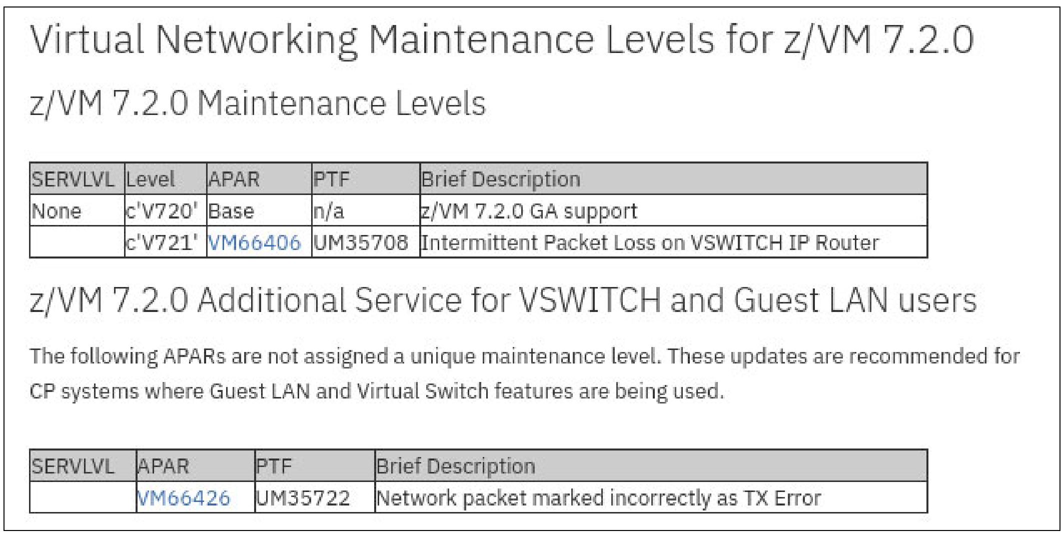
   This query shows that the controller is named `DTCVSW1`.

2. Use the **`NETSTAT`** command with the controller name to determine the maintenance of the `TCPIP MODULE`. This command is stored on disk 592 of the TCPMAINT virtual machine, so you must link to and access it:

```
==> vmlink tcpmaint 592
DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z

==> netstat tcp dtcvsw1 level
```

```
VM TCP/IP Netstat Level 720       TCP/IP Server Name: DTCVSW1

IBM 3907; z/VM Version 7 Release 2.0, service level 2001 (64-bit), VM TCP/IP
Lev
el 720; RSU 0000 running TCPIP MODULE E2 dated 06/24/20 at 13:06
TCP/IP Module Load Address:  00BEF000

Configuration Files in use:
Stack profile file: DTCVSW1 TCPIP E1
DTCPARMS 'server' definition (DTCVSW1) from file: IBM DTCPARMS E1
DTCPARMS 'class' definition (STACK) from file:IBM DTCPARMS E1
```

This command shows information about the current `TCPIP MODULE`. Because we did not apply any service on TCPIP and z/VM uses the base service level (RSU 0000), it is recommended to search the z/VM Virtual Networking CP Maintenance website for any update, as shown in Figure 8-7.



Virtual Networking Maintenance Levels for z/VM 7.2.0

z/VM 7.2.0 Maintenance Levels

SERVLVL	Level	APAR	PTF	Brief Description
None	c'V720'	Base	n/a	z/VM 7.2.0 GA support
	c'V721'	VM66406	UM35708	Intermittent Packet Loss on VSWITCH IP Router

z/VM 7.2.0 Additional Service for VSWITCH and Guest LAN users

The following APARs are not assigned a unique maintenance level. These updates are recommended for CP systems where Guest LAN and Virtual Switch features are being used.

SERVLVL	APAR	PTF	Brief Description
	VM66426	UM35722	Network packet marked incorrectly as TX Error

Figure 8-7 Available service for z/VM virtual networking

Two APARs are available for TCPIP. The procedures to download and apply the service are the same as any PTF: access IBM Software Shopz website, find APARs by number, download them and apply them.

8.6 Moving on to Linux

The installation, configuration, and service of z/VM are complete. With all updates applied, see Chapter 9, “z/VM Centralized Service Management” on page 275 for more information about z/VM management and Linux.

9

z/VM Centralized Service Management

A new method of managing service for sets of stand-alone z/VM systems was introduced with z/VM 7.2. Centralized Service Management (z/VM CSM) allows you to manage distinct levels of service for a specific group of z/VM systems (locally or remotely) from one central system.

This chapter describes the z/VM CSM function. We also describe how to plan for, set up, and use z/VM CSM to manage service across several z/VM systems.

> **Note:** z/VM CSM cannot be used with z/VM SSI. In an SSI cluster, service across the cluster is managed by VMSES/E in the cluster. It is not possible to integrate z/VM CSM with z/VM SSI to deliver service from a system in a z/VM SSI cluster, nor is it possible to use z/VM CSM to deliver service to a z/VM SSI cluster.

z/VM CSM allows maintenance packages to be built, distributed, and applied to up to 55 z/VM systems from a single service management system. Not only does this simplify the task of service distribution, but it also makes it much easier to ensure that systems are kept consistent in service.

For more information about z/VM CSM, see Part 2 of *z/VM 7.2 Service Guide*, GC24-6325.

This chapter includes the following topics:

- 9.1, “z/VM CSM structure” on page 276
- 9.2, “Setting up z/VM CSM” on page 277
- 9.3, “Working with z/VM CSM” on page 283

9.1 z/VM CSM structure

When z/VM CSM is used, one system is designated as the *principal system*. This system is used to build service packages that are delivered to a set of defined *managed systems*. The z/VM CSM principal system builds service levels by using VMSES/E commands.

When a system is participating in z/VM CSM, a dedicated FTP server is set up specifically for transporting service levels and the communication between principal and managed systems.

A new command, **SERVMGR**, is used to manage z/VM CSM. **SERVMGR** uses VMSES/E commands to apply service and local modifications, build serviced content, and drive the transport of z/VM CSM packaged service to managed systems.

z/VM CSM uses Shared File System (SFS) directories to manage service levels for the managed systems. Management of these directories is handled by **SERVMGR**.

9.1.1 z/VM CSM flow overview

The z/VM CSM flow includes the following overall steps:

1. The structures for z/VM CSM are started. On the system that is the principal system, the **SERVMGR** command is run with the **INITIALIZE** operand to set up data structures and establish that system as the z/VM CSM principal system. This process creates the SFS directory structure, and the z/VM CSM service status table.
2. A new level of service that is composted of selected IBM service deliverables or local modifications (or both) is created. The **SERVICE** command is used to build the service content, which is placed in a new SFS structure.
3. Systems to be managed are added to the z/VM CSM management group. Information about the systems that is managed by using z/VM CSM functions is contained in the z/VM CSM system status table.

   > **Note:** After a system is added to a z/VM CSM management group, all the **SERVICE** commands (except **SERVICE ... STATUS**) are blocked on that system. Only z/VM CSM commands can be used to apply service to that system.

4. The built service level is packaged and transported to the managed system or systems by using the dedicated FTP server on each system. Up to 54 managed systems are supported. The z/VM CSM service package is made up of separate SERVLINK files for each z/VM component being serviced.
5. z/VM CSM commands are used to query each managed system for its service processing state (pending, received, in production, or error).

9.1.2 z/VM CSM system requirements

The following requirements must be met for z/VM CSM to be used:

- z/VM CSM and z/VM SSI do not inter-operate. A system in a z/VM SSI cluster cannot be a principal or managed system in z/VM CSM.
- TCP/IP is required on all z/VM CSM systems, both the principal and all managed systems. A dedicated FTP server is used for z/VM CSM management of remote systems. The **CSMSERVE** user ID is defined as part of z/VM 7.2 for you to use for this purpose. For more information about describes the specific steps that are required for configuring the **CSMSERVE** server for use, see Chapter 8 of *z/VM 7.2 Service Guide*, GC24-6325.
- To initialize z/VM CSM, sufficient SFS file pool space on the principal system is needed. The VMPSFS file pool is used for z/VM CSM. To initialize z/VM CSM successfully, at least 3000 4-K blocks of free space are required in Group 1 and 520000 4-K blocks of free space is required in Group 2. In addition, it is highly recommended to have extra space available for new service levels.
- The PTF for APAR VM66428 must be installed on the principal system and all managed systems *before* initializing z/VM CSM. In addition, the PTF must be included as part of any customer-defined service level that is based on the initial z/VM 7.2 RSU.

In addition to the system requirements, some configuration requirements must be met before z/VM CSM is used. For more information about these requirements, see “Overall system configuration changes” in Chapter 8 of *z/VM 7.2 Service Guide*, GC24-6325. They include changes to the TCP/IP configuration to support the **CSMSERVE** FTP server, and ensure that the management user IDs have suitable privilege classes and file pool authority to operate correctly.

9.2 Setting up z/VM CSM

We followed the process as documented in Chapter 9 of *z/VM 7.2 Service Guide*, GC24-6325 to enable z/VM CSM on our systems.

9.2.1 VMPSFS file pool changes

z/VM CSM uses the VMPSFS file pool. Changes must be made to VMSERVP, the service ID (also know as the service virtual machine [SVM]) that hosts VMPSFS to support the requirements of z/VM CSM.

Administrator authority

The MAINTCSM and CSMSERVE user IDs must be configured as permanent administrators of the VMPSFS file pool. We completed this configuration by adding the IDs to the **ADMIN** statement in the DMSPARMS file on the VMSERVP 191 minidisk:

```
ADMIN CSMSERVE MAINTCSM
```

The line can be added after the existing ADMIN lines in the file.

Note: Because the VMSERVP server accesses the disk in read/write mode, it is not possible to access the disk while the file pool server is running. You can wait for a time when the file pool is not in use, and make the changes by logging on to the VMSERVP SVM and stopping the file pool server. We chose to log out of the file pool server by using **`SIGNAL SHUTDOWN`**, and then, connected to the 191 disk by using **`VMLINK VMSERVP 191 (WRITE FILEL`**. From the resulting FILELIST window, we edited VMPSFS DMSPARMS to make the required change. After we exited FILE List, we then logged on the server by using **`XAUTOLOG VMSERVP`**.

Other disk space requirements

Extra space must be added to the file pool. This two-stage process requires that the user ID VMSERVP has disk space added at the directory level, and then, the extra space is defined to the suitable storage group.

Note: For more information about working with file pool servers, see *z/VM Version 7 Release 2 CMS File Pool Planning, Administration, and Operation*, SC24-6261.

The amount of space that is needed depends on the number of different service levels to be maintained for distribution by using z/VM CSM. For our example system, we added two 3390 Model 9 disks to the file pool server by completing the following steps:

1. We obtained two new DASD from available free disk space.
2. The new DASD were formatted and labeled (see “Adding page volumes and perm (user) volumes”, starting from “Formatting DASD for minidisks” on page 191). We labeled the DASD `VMFCS1` and `VMFCS2`.
3. We defined the DASD to DirMaint by modifying `EXTENT CONTROL` (see “Customizing the EXTENT CONTROL file” on page 313).
4. We added the entire space (less cylinder 0) of the new volumes as new minidisks to VMSERVP. We defined the space as minidisks 312 and 313, by using the same parameters as the existing file pool disks (link mode WR, read/write passwords specified).

   Note: We found that the DIRMAINT operation was affected by directory update operations on the VMSERVP ID. At first, we used the **`DIRM AMDISK`** command to add the first of the two minidisks to VMSERVP, but DirMaint became unresponsive after that and we had to recycle the DIRMAINT SVM to add the second minidisk.

5. By using the **`FILEPOOL MINIDISK VMSERVP`** command, we added the disks individually to the running file pool. The output of this command for the first disk is shown in Example 9-1.

   Example 9-1 FILEPOOL MINIDISK command processing

```
filepool minidisk vmservp
DMSWFP3485I FILEPOOL processing begun at 11:14:29 on 10 Oct 2020.
DMSJMD3425R Enter MDK number (nnnnn), virtual device address (vvvv),
DMSJMD3425R and storage group number (ggggg) for a minidisk to be added.
DMSJMD3425R Use format nnnnn vvvv ggggg
00009 0312 2
DMSJMD3426I The following minidisk(s) will be formatted and reserved
DMSJMD3426I for VMSERVP on RDBKZVMF
DMSJMD3426I MDK00009 0312 00002
DMSJMD3427R FORMAT will erase all files on the above minidisk(s).
DMSJMD3427R Do you wish to continue? Enter 1 (Yes) or 0 (No)
```

```
1
DMSJMD3533I Linking to minidisk MDK00009 at 0312 as FFFF.
DASD FFFF DETACHED
DMSJMD3423I The minidisk with virtual device address FFFF has been formatted
VMSERVP : DMS4FM3922I 1 minidisk(s) added to the file pool
DMSJMD3922I 1 minidisk(s) added to the file pool
DMSWFP3486I FILEPOOL processing ended at 11:16:29 on 10 Oct 2020.
```

After the disk was added, we saw a message that indicated that the DIRMAINT SVM was logged off the system. We left DIRMAINT logged off while we added the second disk in the same way. After it was added successfully, we checked the storage group by using **QUERY FILEPOOL**, as shown in Example 9-2.

Example 9-2 Checking the addition of file pool space

```
q filepool storgrp vmpsfs
                       VMPSFS   File Pool Storage Groups

Start-up Date 10/10/20                                   Query Date 10/10/20
Start-up Time 11:04:59                                   Query Time 11:29:23
========================================================================
STORAGE GROUP INFORMATION

 Storage        4K Blocks       4K Blocks
 Group No.        In-Use          Free
     1          4704 -  16%      24059
     2        345894 -   8%    3885390
========================================================================
```

At this point, the file pool operation was complete.

The objective of the use of online file pool operation commands was to eliminate disruption to file pool users. Because the file pool had to be shut down for the DMSPARMS file update, and the file pool change appeared to affect DirMaint, this process was not successful.

An alternative process for adding file pool space, which requires the file pool to be shut down, is described in Chapter 9, “Adding Minidisks in Dedicated Maintenance Mode” of *z/VM: CMS File Pool Planning, Administration, and Operation*.

9.2.2 User ID privilege class

The MAINTCSM user ID requires access to the **FORCE** command. Because the **FORCE** command is part of system privilege class A, class A was added to MAINTCSM as described in *z/VM 7.2 Service Guide*, GC24-6325.

9.2.3 TCP/IP configuration changes

The TCP/IP configuration of your systems must be updated to support z/VM CSM.

DTCPARMS entry

An entry in the `SYSTEM DTCPARMS` file is needed for the CSMSERVE to operate correctly. We edited this file and added the entry, as shown in Example 9-3.

Example 9-3 SYSTEM DTCPARMS entry for the CSMSERVE server

```
:nick.CSMSERVE :type.server :class.ftp
               :parms.CSMSERVE CONFIG *
```

PROFILE TCP/IP changes

The following changes are required to the `PROFILE TCPIP` for your system:

- AUTOLOG Statement: An entry for the CSMSERVE ID is added to AUTOLOG.
- PORT Statement: Port reservations for the CSMSERVE ID are added to PORT.
- Connection limit: The connection limit for foreign IP addresses is set.

The required changes are shown in Example 9-4.

Example 9-4 PROFILE TCP/IP examples

```
. . .
AUTOLOG
  . . .
  CSMSERVE 0
ENDAUTOLOG
. . .
PORT
  . . .
  4534 TCP CSMSERVE  NOAUTOLOG ; z/VM CSM FTP Server (Data)
  4535 TCP FTPSERVE            ; z/VM CSM FTP Server (control)
. . .
FOREIGNIPCONLIMIT 200
```

CSMSERVE CONFIG

The configuration file for the VMCSM FTP server is supplied as a sample, which must be copied. The sample is `CSMSERVE SCONFIG`, on TCPMAINT 591, and it must be copied to `CSMSERVE CONFIG` on TCPMAINT 198. If needed, modify `CSMSERVE CONFIG` to include parameters that are required for your site (such as encryption).

If changes are made to `CSMSERVE CONFIG`, it is recommended not to change the KEEPALIVE and INACTIVE statements. These settings were made to support the operation of VMCSM, and are changed only after consulting with IBM Support Center.

FTP DATA update on principal system

The default value of DataCtTime must be changed on the system that is the VMCSM principal system. Copy the IBM supplied `FTP SDATA` file from TCPMAINT 592 to MAINTCSM 191 as `FTP DATA`. Then, in the copied file, set the DataCtTime value to 720 seconds:

```
DataCtTime 720
```

For each software release that is maintained by VMCSM, this file must be copied to the CSM root directory for that release.

9.2.4 VMCSM APAR installation

The PTF for APAR VM66428 must be installed on all systems that are participating in VMCSM.

Example 9-5 shows the listing of the MAINT720 500 disk after we uploaded the maintenance envelope (the file had a long name when downloaded from IBM, but we uploaded it as `VMCSM TERSE`).

Example 9-5 FILELIST view of uploaded PTF file

```
MAINT720 FILELIST A0  V 169  Trunc=169 Size=2 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      VMCSM    TERSE    F1 V      32768         22        174 10/10/20  9:18:37
      7201RSU1 SERVLINK F1 V      65535     363866      16384  8/05/20  9:42:48
```

We used the **DETERSE** command to extract the file (in the FILELIST view next to `VMCSM TERSE`, we entered **DETERSE / = SERVLINK =**). When we refreshed the FILELIST view, we saw the correctly extracted file, as shown in Example 9-6.

Example 9-6 Extracted PTF file in the filelist view

```
MAINT720 FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=14
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      VMCSM    SERVLINK F1 V       4005        558        472 10/10/20  9:29:53
      VMCSM    TERSE    F1 V      32768         22        174 10/10/20  9:18:37
      7201RSU1 SERVLINK F1 V      65535     363866      16384  8/05/20  9:42:48
```

Then, we used the **SERVICE** command to install the PTF. Example 9-7 shows the result of this **SERVICE** run (only the last few messages are shown).

Example 9-7 Initial SERVICE output from VMCSM PTF apply

```
. . .
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
VMFSET2760I VMFSETUP processing started for DETACH VMSESSFS
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFSUI2760W VMFSUFIN processing incomplete due to service that affects core
            VMSES/E parts
VMFSUI1211I A Checkpoint Restart Record has been created for package VMCSM
            in the System-Level Restart Table
VMFSRV2310W Program restart file, SERVICE $RESTART, has been created or
            persists due to service that affects core VMSES/E parts. Restart
            SERVICE to resume processing, using the command that follows:
            SERVICE RESTART
VMFSRV2264I Restoring prior system environment using saved access/minidisk
            information
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            SERVICEEXEC20201010093437
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFSRV2760W SERVICE processing incomplete due to service that affects core
            VMSES/E parts
```

Because the service affects core VMSES/E components, it cannot be completed in a single pass. We issued **SERVICE RESTART** as instructed, and processing resumed, as shown in Example 9-8 (again, not all messages shown).

Example 9-8 Completing the VMCSM SERVICE run

```
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFSRV2195I SERVICE RESTART
VMFSRV2263I Saving access/minidisk information for existing system
            environment
VMFSRV1242I File SERVICE $RESTART D exists; Content is:
VMFSRV1242I SESRESET ALL
VMFSRV2760I SERVICE RESTART processing started
VMFSRV2263I Saving access/minidisk information for existing system
            environment
VMFSET2760I VMFSETUP processing started for ENVSAVE
            SERVICEEXEC20201010093509
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
. . .
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFUTL2204I Linking PMAINT 41D with link mode M
VMFUTL2204I Linking PMAINT 41D with link mode M
VMFSRV1233I The following products have been serviced.
VMFSRV1233I VMSESSFS
VMFSRV2264I Restoring prior system environment using saved access/minidisk
            information
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            SERVICEEXEC20201010093509
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFSRV2760I SERVICE processing completed successfully (RC=0)
```

We then ran **PUT2PROD**. Selected output is shown in Example 9-9.

Example 9-9 PUT2PROD output for the VMCSM PTF

```
. . .
VMFP2P2760I PUT2PROD processing started for COPYPART
VMFP2P2204I Linking MAINT 190 as 0190 with link mode M
DASD 0190 LINKED R/W; R/O BY    16 USERS
DMSACP725I 190 also = S disk
VMFP2P2204I Linking MAINT 19D as 019D with link mode M
DASD 019D LINKED R/W; R/O BY    15 USERS
VMFP2P1231I Copying files from MAINT720 49D to MAINT 19D
VMFP2P2204I Linking MAINT 402 as 0402 with link mode M
DASD 0402 LINKED R/W; R/O BY    13 USERS
VMFP2P1231I Copying files from MAINT720 49D to MAINT 402
VMFP2P1231I Copying files from MAINT720 490 to MAINT 190
DMSACP726I 5E6 B released
VMFP2P1231I Copying files from MAINT720 5E6 to MAINT720 5E5
. . .
HCPNSD440I The Named Saved System (NSS) CMS was successfully defined in
HCPNSD440I fileid 0047.
BLDCMS  : HCPNSS440I Named Saved System (NSS) CMS was successfully saved in
BLDCMS  : HCPNSS440I fileid 0047.
```

```
BLDCMS   : z/VM V7.2.0    2020-10-07 10:05
HCPQCS150A User BLDCMS has issued a VM read
HCPNSD440I The Named Saved System (NSS) ZCMS was successfully defined in
HCPNSD440I fileid 0048.
BLDCMS   : Ready; T=0.01/0.01 23:07:32
BLDCMS   : HCPNSS440I Named Saved System (NSS) ZCMS was successfully saved in
BLDCMS   : HCPNSS440I fileid 0048.
BLDCMS   : z/CMS V7.2.0   2020-10-07 10:05
HCPQCS150A User BLDCMS has issued a VM read
. . .
VMFP2P1233I The following products have been put into production. Recycle
            the appropriate servers.
VMFP2P1233I VMSES
VMFP2P2264I Restoring prior system environment using saved access/minidisk
            information
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            PUT2PRODEXEC20201010095021
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFP2P2760I PUT2PROD processing completed successfully (RC=0)
```

Files were copied to the CMS and VMSES/E disks, and then, the CMS Named Saved Systems were saved.

9.3 Working with z/VM CSM

After the setup work is complete, you are ready to initialize VMCSM and start building service releases.

9.3.1 Initializing z/VM CSM by using SERVMGR INIT

We logged on to our principal system with the MAINTCSM ID. We ensured the `FTP DATA` change that is described in "FTP DATA update on principal system" was done. Then, we ran the **SERVMGR** command to initialize VMCSM. Example 9-10 shows the result of running this command.

Example 9-10 SERVMGR INIT 720 output

```
servmgr init 720
VMFCMG2195I SERVMGR INIT 720
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSY2760I CSMSYSMT processing started
VMFCSY2195I CSMSYSMT CMG
VMFCSY2936I FILEPOOL RELOAD command processing is in progress, which might
            require several seconds (or minutes) to complete
VMFCSY4042I Service level 720.BASE created
VMFCSY4003I File CSM SVCSTAT created
VMFCTL3040I Issuing command VMFBTMAP with operands/options: ICKDSFSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP ICKDSFSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: LESFS (LOG
            CMG+$CSM RETAIN X
```

```
VMFCTL3038I Command VMFBTMAP LESFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: AVSSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP AVSSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: CMSSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP CMSSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: CPSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP CPSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: DIRMSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP DIRMSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: DVSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP DVSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: GCSSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP GCSSFS (LOG CMG+$CSM RETAIN X completed with
RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: VMHCDSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP VMHCDSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: PERFTKSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP PERFTKSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: RACFSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP RACFSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: REXXSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP REXXSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: RSCSSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP RSCSSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: VMSESSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP VMSESSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: TCPIPSFS (LOG
            CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP TCPIPSFS (LOG CMG+$CSM RETAIN X completed with
            RC=0
VMFCTL3040I Issuing command VMFBTMAP with operands/options: TSAFSFS (LOG
```

```
          CMG+$CSM RETAIN X
VMFCTL3038I Command VMFBTMAP TSAFSFS (LOG CMG+$CSM RETAIN X completed with
          RC=0
VMFCTL4003I File PRODID BITMCSM$ X created
VMFCTL3038I Command VMFSETUP CSM ICKDSFSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM LESFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM AVSSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM CMSSFS (NOCONS LOG CMG+$CSM RETAIN X
completed with RC=0
VMFCTL3038I Command VMFSETUP CSM CPSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM DIRMSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM DVSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM GCSSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM VMHCDSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
DASD 0201 LINKED R/W; R/O BY PERSMAPI
VMFSET2204I Linking PERFSVM 201 as 201 with link mode MR
VMFCTL3038I Command VMFSETUP CSM PERFTKSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM RACFSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM REXXSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM RSCSSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM VMSESSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM TCPIPSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL3038I Command VMFSETUP CSM TSAFSFS (NOCONS LOG CMG+$CSM RETAIN X
          completed with RC=0
VMFCTL4003I File PRODID VVTLCL X created
VMFCSY2760I CSMSYSMT processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
```

VMCSM is now initialized. A new `VMPSFS:CSM720` directory was created that contains the CSM structures for z/VM 7.2.

The customized `FTP DATA` file now must be copied from the MAINTCSM 191 disk to the newly created `VMPSFS:CSM720` directory. Because SERVMGR accessed `VMPSFS:CSM720` as filemode A, and MAINTCSM 191 is now accessed as filemode Z, the following command to copy the file is used:

```
COPYF FTP DATA Z = = A (OLDD
```

9.3.2 Creating a service level

To start managing service on z/VM systems by using VMCSM, service levels must be created. The first service level contains all of the service that was shipped with the base z/VM product (further service is added to it before it is delivered to managed systems).

We used the **SERVMGR SRVLVL ADD** command to create a service level, as shown in Example 9-11.

Example 9-11 Creating a service level by using SERVMGR

```
servmgr srvlvl add 720 rsu1 based on base nonesm desc RSU level shipped with prod
uct
VMFCMG2195I SERVMGR SRVLVL ADD 720 RSU1 BASEDON BASE NONESM DESC RSU LEVEL
            SHIPPED WITH PRODUCT
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSV2760I CSMSRVMT EXEC processing started
VMFCSV2195I CSMSRVMT SRVLVL ADD 720 RSU1 BASEDON BASE NONESM DESC RSU LEVEL
            SHIPPED WITH PRODUCT
VMFCSV4115I Creating 138 directories for new service level RSU1
VMFCSV4115I Created 10 of 138 directories for new service level RSU1
VMFCSV4115I Created 20 of 138 directories for new service level RSU1
VMFCSV4115I Created 30 of 138 directories for new service level RSU1
VMFCSV4115I Created 40 of 138 directories for new service level RSU1
VMFCSV4115I Created 50 of 138 directories for new service level RSU1
VMFCSV4115I Created 60 of 138 directories for new service level RSU1
VMFCSV4115I Created 70 of 138 directories for new service level RSU1
VMFCSV4115I Created 80 of 138 directories for new service level RSU1
VMFCSV4115I Created 90 of 138 directories for new service level RSU1
VMFCSV4115I Created 100 of 138 directories for new service level RSU1
VMFCSV4115I Created 110 of 138 directories for new service level RSU1
VMFCSV4115I Created 120 of 138 directories for new service level RSU1
VMFCSV4115I Created 130 of 138 directories for new service level RSU1
VMFCSV4116I Updated VMSESE PROFILE to reflect new service level RSU1
VMFCSV4116I Created new CSM PPF to reflect new service level RSU1
VMFCSV4116I Created specific VMFINS DEFAULTS file for service level RSU1
VMFCSV4116I Updated CSM SVCSTAT for new service level RSU1
VMFCSV4116I Updated SERVLVL DESCRIPT on VMPSFS:CSM720.RSU1 to reflect new
            service level RSU1
VMFCSV4042I Service level RSU1 created
VMFCSV2760I CSMSRVMT EXEC processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
```

The command we issued performed the following tasks:

- Created a service level in version 7.2.0 called `RSU1`. We called it RSU1 because after it is created, we add the content of the first RSU.
- Based the service level on the supplied maintenance level, `BASE`.
- Defined the service level to not include an External Security Manager - `NONESM`.
- Provided the text `RSU level shipped with product` as a description for the service level.

Next, we added the RSU level that was applied to the system at installation time by using the **SERVMGR SRVLVL SERVICE** command. We needed to access the MAINT720 500 disk where the RSU deliverable is stored. Then, we ran the **SERVMGR** command to include the RSU into our VMCSM service level. The steps are shown in Example 9-12.

Example 9-12 Adding the RSU to the VMCSM service level

```
link maint720 500 500 rr
DASD 0500 LINKED R/O; R/W BY MAINT720
Ready;
acc 500 c
DMSACP723I C (500) R/O
Ready;
servmgr srvlvl service 720 rsu1 all 7201rsu1
VMFCMG2760I SERVMGR processing started
. . .
< < Many SERVICE command messages appear > >
. . .
VMFSRV2760I SERVICE processing completed successfully (RC=0)
VMFCSV2760I SERVICE ALL 7201RSU1 ( PPFNAME CSM processing completed
            successfully (RC=0)
VMFCTL3040I Issuing command VMFBTMAP with operands/options: LESFS (LOG
            CMG+$CSM RETAIN T
VMFCTL3038I Command VMFBTMAP LESFS (LOG CMG+$CSM RETAIN T completed with
            RC=0
VMFCTL4003I File PRODID BITMCSM$ T created
VMFCTL3040I Issuing command VMFBTMAP with operands/options: CPSFS (LOG
            CMG+$CSM RETAIN T
VMFCTL3038I Command VMFBTMAP CPSFS (LOG CMG+$CSM RETAIN T completed with
            RC=0
VMFCTL4003I File PRODID BITMCSM$ T created
VMFCSV4116I Updated CSM SVCSTAT for new service level RSU1
VMFCSV2760I CSMSRVMT EXEC processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

The **SERVMGR** command that we ran selected the service level `RSU1` in version `720` and performed the **SERVICE** command against that service level by using the parameters **ALL 7201RSU1**.

The messages up to and including `VMFSRV2760I` are mostly the same as was observed during installation when the system performed **SERVICE ALL 7201RSU1**. More messages after that point are issued by VMCSM to indicate the actions on our service level.

The **VMFVIEW** command is available for viewing VMCSM messages in the same way as other VMSES/E messages can be viewed. After the **SERVMGR SRVLVL SERVICE** command was used, we ran **VMFVIEW CSM** to show the rest of the messages (we pressed **PF2** to show all messages. The output is shown in Example 9-13.

Example 9-13 VMFVIEW CSM after SERVMGR SRVLVL SERVICE

```
===> VMFVIEW - Message Log Browse of $CSMCMG  $MSGLOG  A1 <===
You are viewing ALL ¬TR: messages from the LAST run.
Number of messages shown = 75 <===> Number of messages not shown = 0
**********************************************************************
****      SERVMGR      SYSTEM: RDBKZVMF      USERID: MAINTCSM      ****
**********************************************************************
```

```
****          Date: 2020-10-11         Time: 23:27:56            ****
********************************************************************
ST:VMFCMG2195I SERVMGR SRVLVL SERVICE 720 RSU1 ALL 7201RSU1
ST:VMFCMG2760I SERVMGR processing started
ST:VMFUTL2767I Reading VMSESE PROFILE B for additional options
ST:VMFCSV2760I CSMSRVMT EXEC processing started
ST:VMFCSV2195I CSMSRVMT SRVLVL SERVICE 720 RSU1 ALL 7201RSU1
ST:VMFCSV2760I SERVICE ALL 7201RSU1 ( PPFNAME CSM processing started
ST:VMFCSV2760I SERVICE ALL 7201RSU1 ( PPFNAME CSM processing completed
ST:            successfully (RC=0)
ST:VMFSET2760I VMFSETUP processing started for ENVSAVE
ST:            SCSMUTLSEXEC20201011232807
ST:VMFSET2760I VMFSETUP processing completed successfully (RC=0)
ST:VMFCTL3040I Issuing command VMFBTMAP with operands/options: LESFS (LOG
ST:            CMG+$CSM RETAIN T
ST:VMFBMP2760I VMFBTMAP processing started
ST:VMFBMP2760I VMFBTMAP processing started for CSM LESFS
ST:VMFSET2760I VMFSETUP processing started for CSM LESFS
ST:VMFUTL2205I Minidisk|Directory Assignments:
ST:            String    Mode  Stat  Vdev  Label  (OwnerID Odev : Cyl/%Used)
ST:                                        -or-       SFS Directory Name
ST:VMFUTL2205I LOCALMOD  E/E   R/O   DIR   VMPSFS:CSM720.RSU1.LE.LOCALMOD
ST:VMFUTL2205I LOCALSAM  F/F   R/O   DIR   VMPSFS:CSM720.RSU1.LE.SAMPLE
ST:VMFUTL2205I APPLY     G/G   R/O   DIR   VMPSFS:CSM720.RSU1.LE.APPLYALT
ST:VMFUTL2205I           H/H   R/O   DIR   VMPSFS:CSM720.RSU1.LE.APPLYINT
ST:VMFUTL2205I           I/I   R/O   DIR   VMPSFS:CSM720.RSU1.LE.APPLYPROD
ST:VMFUTL2205I DELTA     J/J   R/O   DIR   VMPSFS:CSM720.RSU1.LE.DELTAPROD
ST:VMFUTL2205I BUILD0    K/K   R/O   DIR   VMPSFS:CSM720.RSU1.LE.49E
ST:VMFUTL2205I BUILD2    L/L   R/O   DIR   VMPSFS:CSM720.RSU1.LE.49B
ST:VMFUTL2205I BUILD4    M/M   R/O   DIR   VMPSFS:CSM720.RSU1.LE.4DD
ST:VMFUTL2205I BASE1     N/N   R/O   DIR   VMPSFS:CSM720.RSU1.LE.OBJECT
ST:VMFUTL2205I --------  A     R/W   DIR   VMPSFS:CSM720.
ST:VMFUTL2205I --------  B     R/W   DIR   VMPSFS:CSM720.RSU1.VMSES.5E6
ST:VMFUTL2205I --------  C     R/O   500   MNT500 (MAINT720 0500 : 1200/08)
 1=Help       2=All        3=Quit       4=Exception  5=Status     6=Build
 7=Backward   8=Forward    9=OutCompRq 10=Non-Stat  11=Requisite 12=Severe
====>
```

At the bottom of Example 9-13 on page 287, you can see the `VMFUTL2205I` messages that show the directory assignments for that stage of the service process. You also can see all of the standard VMSES/E aliases, such as APPLY, DELTA, LOCALMOD, and BUILDx. This result shows part of how VMCSM works: it creates an “image” of what an installed system looks like within directories in VMPSFS, and then applies service to that image in the same way that service is applied to a live system.

However, before the service level can be used for a system, we also must include the VMCSM PTF to the service level by using following command:

```
servmgr srvlvl service 720 rsu1 all vmcsm
```

We found that the PTF did not complete in a single run and required a restart (just as it did on the live systems). Our result is shown in Example 9-14.

Example 9-14 Initial PTF apply run under VMCSM

```
servmgr srvlvl service 720 rsu1 all vmcsm
. . .
VMFSUI2760W VMFSUFIN processing incomplete due to service that affects core
            VMSES/E parts
VMFSUI1211I A Checkpoint Restart Record has been created for package VMCSM
            in the System-Level Restart Table
VMFSRV2310W Program restart file, SERVICE $RESTART, has been created or
            persists due to service that affects core VMSES/E parts. Restart
            SERVICE to resume processing, using the command that follows:
            SERVICE RESTART
VMFSRV2264I Restoring prior system environment using saved access/minidisk
            information
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            SERVICEEXEC20201011235625
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFSRV2760W SERVICE processing incomplete due to service that affects core
            VMSES/E parts
VMFCSV2760W SERVICE ALL VMCSM ( PPFNAME CSM processing incomplete due to
            service that affects core VMSES/E parts
VMFCSV4068I Review message log $VMFSRV on VMPSFS:CSM720. for details
VMFCSV2760E CSMSRVMT EXEC processing completed unsuccessfully (RC=5)
VMFCMG1965E Command SERVMGR failed with RC=5 when issued with argument(s):
            SRVLVL SERVICE 720 RSU1 ALL VMCSM
VMFCMG2760E SERVMGR processing completed unsuccessfully (RC=5)
Ready(00005);
```

We tried the **SERVMGR SRVLVL SERVICE 720 RSU1 RESTART** command, but SERVMGR passed the incorrect parameters to SERVICE. Checking the HELP for SERVMGR, we found that RESTART is an option for the command. We retried with **SERVMGR SRVLVL SERVICE 720 RSU1 (RESTART** and were successful, as shown in Example 9-15.

Example 9-15 Successful PTF apply under VMCSM

```
servmgr srvlvl service 720 rsu1 (restart
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSV2760I CSMSRVMT EXEC processing started
VMFCSV2195I CSMSRVMT SRVLVL SERVICE 720 RSU1 ( RESTART
VMFCSV2760I SERVICE RESTART ( PPFNAME CSM processing started
VMFCSV2760I SERVICE RESTART ( PPFNAME CSM processing completed
            successfully (RC=0)
VMFSET2760I VMFSETUP processing started for ENVSAVE
            SCSMUTLSEXEC20201012000332
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
. . .
VMFBMP1909I VMPFX-UM BITMAP created on your D-disk
VMFBMP1909I VMPRODS BITMAP created on your D-disk
VMFBMP1909I VMPFXALL BITMAP created on your D-disk
VMFCTL3038I Command VMFBTMAP VMSESSFS (LOG CMG+$CSM RETAIN T completed
            with RC=0
VMFCTL4003I File PRODID BITMCSM$ T created
```

```
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            SCSMUTLSEXEC20201012000353
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFCSV4116I Updated CSM SVCSTAT for new service level RSU1
VMFCSV2760I CSMSRVMT EXEC processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

Now our service level was successfully created, and we can query information about the service level. Example 9-16 shows the results of the **SERVMGR SRVLVL QUERY** command on our system.

Example 9-16 VMCSM command SERVMGR SRVlvl Query

```
servmgr srvlvl query 720 all
VMFCMG2195I SERVMGR SRVLVL QUERY 720 ALL
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SRVLVL Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4200I Service Level: BASE
VMFCSQ4200I Root Directory: VMPSFS:CSM720.
VMFCSQ4200I Service Level added on: 10/10/20
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 10/10/20
VMFCSQ4200I ESM Enablement: NO
VMFCSQ4200I Service Level build state: STABLE
VMFCSQ4200I Service Level update lock: Yes
VMFCSQ4200I Description:
VMFCSQ4200I  Service level: BASE     Created by: MAINTCSM 10/10/20 12:25:02
VMFCSQ4200I  Based on: BASE
VMFCSQ4200I  Description: z/VM 7.2.0 Base Service Level
VMFCSQ4200I
VMFCSQ4200I Service Level: RSU1
VMFCSQ4200I Root Directory: VMPSFS:CSM720.
VMFCSQ4200I Service Level added on: 10/11/20
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 10/12/20
VMFCSQ4200I ESM Enablement: NO
VMFCSQ4200I Service Level build state: TEST
VMFCSQ4200I Service Level update lock: No
VMFCSQ4200I Description:
VMFCSQ4200I  Service level: RSU1    Created by: MAINTCSM 10/11/20 23:02:01
VMFCSQ4200I  Based on: BASE
VMFCSQ4200I  Description: RSU LEVEL SHIPPED WITH PRODUCT
VMFCSQ4206I Full service level description available in SERVLVL DESCRIPT
            file in directory: VMPSFS:CSM720.RSU1
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

We can see the RSU1 service level that we created, which was based on the BASE level. The display also shows us the BASE level. We also can use the DETAILS option against a service level to list all PTFs that are included of the components in that level.

Now, we were ready to add systems to VMCSM management, as described next.

9.3.3 Adding a managed system

We started by managing a single remote system, RDBKZVMC. This system was upgraded from z/VM 7.1 by using Upgrade In Place, and had the PTF for VMCSM applied. We went through the TCP/IP configuration process described in "TCP/IP configuration changes" on page 279, except for the `FTP DATA` file change, which is intended for the principal system only.

We also defined the CSMSERVE ID to be an administrator of the VMPSFS file pool. For our first test, we made this definition temporarily by using the following command:

```
ENROLL ADMINISTRATOR CSMSERVE VMPSFS:
```

We used the **SERVMGR SYSTEM ADD** command to set up RDBKZVMC for VMCSM management, as shown in Example 9-17. The command prompts for a user ID and password to use to authenticate. A special ID of `CSMWORK` is defined in the base system for this purpose.

Example 9-17 Adding a VMCSM managed system, attempt 1

```
servmgr system add 720 rdbkzvmc rsu1 nonesm hostname 9.76.61.239 commtype ftp
VMFCMG2195I SERVMGR SYSTEM ADD 720 RDBKZVMC RSU1 NONESM HOSTNAME 9.76.61.239
            COMMTYPE FTP
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSY2760I CSMSYSMT processing started
VMFCSY2195I CSMSYSMT CMG
VMFCTL4017R Enter the login user ID to be used for RDBKZVMC host system:
            9.76.61.239
csmwork
VMFCTL4018R Enter the login password for user 'csmwork' on RDBKZVMC host
            system: 9.76.61.239
< < password > >
VMFCSY4047I System directory VMPSFS:MAINTCSM.RDBKZVMC created
VMFCTL1965E Command FTP 9.76.61.239 4535 failed (00:45:20) with RC=34539
            when issued with argument(s): (WIDTH 110 NONETRC EXIT
VMFCSY4060E Unable to XAUTOLOG MAINT720 on system RDBKZVMC; XAUTOLOG return
            code 54
VMFCSY4046E Error occurred during SYSTEM ADD processing; completed work will
            be undone
VMFCSY2760E CSMSYSMT processing completed unsuccessfully (RC=8)
VMFCMG1965E Command SERVMGR failed with RC=8 when issued with argument(s):
            SYSTEM ADD 720 RDBKZVMC RSU1 NONESM HOSTNAME 9.76.61.239
            COMMTYPE FTP
VMFCMG2760E SERVMGR processing completed unsuccessfully (RC=8)
Ready(00008);
```

When we first tried this process, we discovered that one of our team used the MAINT720 ID on the system that we wanted to manage by using VMCSM. If this issue occurs, it is important to *not* just log off from the ID at the managed system. If service actions were underway on the to-be-managed system, those actions might bring the system to a state that is inconsistent with the maintenance to be managed by using VMCSM.

We verified that no service activities were occurring on the to-be-managed system; then, retried the command.

> **Note:** We found another configuration requirement that at the time of this writing was not documented in *z/VM 7.2 Service Guide*, GC24-6325. We found that the **SERVMGR SYSTEM ADD** command timed out without completing because the MAINT720 ID was attempting to issue a command to the CSMSERVE ID by using **SMSG** and it was not authorized to do so.
>
> On a z/VM 7.2 fresh installation, we saw that MAINT720 was listed in the OBEY section of PROFILE TCPIP. However, because our candidate system was upgraded from z/VM 7.1, \ no OBEY entry existed for MAINT720.
>
> After MAINT720 was added to the OBEY list in PROFILE TCPIP, the **SERVMGR SYSTEM ADD** command completed as expected.

Example 9-18 shows the result of our successful **SERVMGR SYSTEM ADD** command.

Example 9-18 SERVMGR SYStem ADD command output

```
servmgr system add 720 rdbkzvmc rsu1 nonesm hostname 9.76.61.239 commtype ftp
VMFCMG2195I SERVMGR SYSTEM ADD 720 RDBKZVMC RSU1 NONESM HOSTNAME 9.76.61.239
            COMMTYPE FTP
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSY2760I CSMSYSMT processing started
VMFCSY2195I CSMSYSMT CMG
VMFCTL4017R Enter the login user ID to be used for RDBKZVMC host system:
            9.76.61.239
csmwork
VMFCTL4018R Enter the login password for user 'csmwork' on RDBKZVMC host
            system: 9.76.61.239
< < password > >
VMFCSY4048I System name RDBKZVMC added to CSM
VMFCSY4056W System RDBKZVMC has been added with service level RSU1 pending
            Use the SERVMGR SRVLVL SEND command to send service level RSU1
            to system RDBKZVMC
VMFCSY2760W CSMSYSMT processing completed with warnings (RC=4)
VMFCMG1966W Command SERVMGR ended with RC=4 when issued with argument(s):
            SYSTEM ADD 720 RDBKZVMC RSU1 NONESM HOSTNAME 9.76.61.239
            COMMTYPE FTP
VMFCMG2760W SERVMGR processing completed with warnings (RC=4)
Ready(00004);
```

Reviewing the console of the candidate system, we saw that the MAINT720 user ID was logged on by CSMSERVE and logged off shortly afterward. During that time, VMCSM commands were run on the candidate system to create VMCSM structures and control files. After the command completed, we logged on to MAINT720 on the managed system and reviewed the console files in the reader to observe the VMCSM activity.

The output of the **SERVMGR SYStem ADD** command in Example 9-18 shows that the command ended with a warning. **SERVMGR** reminds us that the RSU1 service level that we assigned the system to was delivered to the system.

After RDBKZVMC was added as a managed system, we issued some query commands to check the status. First, we used **SERVMGR SYSTEM QUERY** to check the status of the systems.

We repeated the display by using the **DETAILS** option against RDBKZVMC to see the extra information that was provided. We also performed another display of the RSU1 service level. Example 9-19 shows the results of these commands.

Example 9-19 VMCSM system query commands

```
servmgr system query 720 all
VMFCMG2195I SERVMGR SYSTEM QUERY 720 ALL
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4202I System Information Details for System: RDBKZVMC
VMFCSQ4202I ESM Enablement: NO
VMFCSQ4202I Current Service Level: <None>
VMFCSQ4202I Pending Service Level: RSU1
VMFCSQ4202I Pending Service Level Status: INSTALLED
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
servmgr system query 720 rdbkzvmc (details
VMFCMG2195I SERVMGR SYSTEM QUERY 720 RDBKZVMC (DETAILS
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4202I System Information Details for System: RDBKZVMC
VMFCSQ4202I VRM: 720
VMFCSQ4202I Communication Protocol: FTP
VMFCSQ4202I Host Name: 9.76.61.239
VMFCSQ4202I FTP Port: 4535
VMFCSQ4202I FTP Command Operands: <None>
VMFCSQ4202I FTP Data File: <None>
VMFCSQ4202I ESM Enablement: NO
VMFCSQ4202I Current Service Level: <None>
VMFCSQ4202I Pending Service Level: RSU1
VMFCSQ4202I Pending Service Level Status: INSTALLED
VMFCSQ4209W System has no current service level
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
servmgr srvlvl query 720 rsu1 (systems
VMFCMG2195I SERVMGR SRVLVL QUERY 720 RSU1 (SYSTEMS
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SRVLVL Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4200I Service Level: RSU1
```

```
VMFCSQ4200I Root Directory: VMPSFS:CSM720.
VMFCSQ4200I Service Level added on: 10/11/20
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 10/12/20
VMFCSQ4200I ESM Enablement: NO
VMFCSQ4200I Service Level build state: TEST
VMFCSQ4200I Service Level update lock: No
VMFCSQ4200I Description:
VMFCSQ4200I  Service level: RSU1    Created by: MAINTCSM 10/11/20 23:02:01
VMFCSQ4200I  Based on: BASE
VMFCSQ4200I  Description: RSU LEVEL SHIPPED WITH PRODUCT
VMFCSQ4206I Full service level description available in SERVLVL DESCRIPT
            file in directory: VMPSFS:CSM720.RSU1
VMFCSQ4200I
VMFCSQ4201I Applicable Systems:
VMFCSQ4201I Current: <None>
VMFCSQ4201I Pending: RDBKZVMC
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

As we saw with the RC=4 when we added RDBKZVMC to VMCSM management, the display of the RSU1 service level reminds us that the service level was delivered to the applicable system RDBKZVMC.

9.3.4 Building a service package

The service level represents a collection of service that are to be applied to managed systems. To \perform this service delivery, a *package* must be created. The package consists of a SERVLINK file for each component that was serviced in the service level. The SERVLINK files that are created are special VMCSM files that containing more information that is needed to update the service inventory files on the managed systems.

The **SERVMGR SRVLVL PACKAGE** command is used to create the package for a service level. We created a package for our RSU1 service level, as shown in Example 9-20.

Example 9-20 SERVMGR SRVLVLPACKAGE command output

```
servmgr srvlvl package 720 rsu1
VMFCMG2195I SERVMGR SRVLVL PACKAGE 720 RSU1
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSV2760I CSMSRVMT EXEC processing started
VMFCSV2195I CSMSRVMT SRVLVL PACKAGE 720 RSU1
VMFCPB2760I CSMPKGBD processing started
VMFCPB2195I
VMFCPB3004I Accessing required minidisks/directories...
VMFCPB4025I Confirming required VMSES/E files are available
VMFCPB4027I Initiating package build for product ID 7VMSES20 (Package 1 of
            3)
VMFCPB4026I Identifying LOCALMOD content for envelope inclusion
VMFCPB4026I Identifying DELTA content for envelope inclusion
VMFCPB4024I Creating file: CSMSES01 ENVELOPE C
VMFCPB4003I File CSMSES01 ENVELOPE C created
```

```
VMFCPB4024I Creating file: CSMSES02 ENVELOPE C
VMFCPB4003I File CSMSES02 ENVELOPE C created
VMFCPB4024I Creating file: CSMSES03 ENVELOPE C
VMFCPB4003I File CSMSES03 ENVELOPE C created
VMFCPB4026I Identifying TOOLS content for envelope inclusion
VMFCPB4026I Identifying SYSTEM content for envelope inclusion
VMFCPB4026I Identifying TASK content for envelope inclusion
VMFCPB4026I Identifying NCHELP content for envelope inclusion
VMFCPB4025I Confirming required 7VMSES20 files are available
VMFCPB4024I Creating file: 7VMSES20 SERVLINK C
VMFCPB2936I VMFPLCD command processing is in progress, which might require
            several seconds (or minutes) to complete
VMFCPB4003I File 7VMSES20 SERVLINK C created
VMFCPB4036I Package build for PRODID 7VMSES20 completed successfully
VMFCPB4027I Initiating package build for product ID 7VMCPR20 (Package 2 of
            3)
VMFCPB4026I Identifying LOCALMOD content for envelope inclusion
VMFCPB4026I Identifying DELTA content for envelope inclusion
VMFCPB4024I Creating file: CSMCPR01 ENVELOPE C
VMFCPB4003I File CSMCPR01 ENVELOPE C created
VMFCPB4024I Creating file: CSMCPR02 ENVELOPE C
VMFCPB4003I File CSMCPR02 ENVELOPE C created
VMFCPB4024I Creating file: CSMCPR03 ENVELOPE C
VMFCPB4003I File CSMCPR03 ENVELOPE C created
VMFCPB4026I Identifying TOOLS content for envelope inclusion
VMFCPB4026I Identifying SYSTEM content for envelope inclusion
VMFCPB4026I Identifying SYSTEMCM content for envelope inclusion
VMFCPB4026I Identifying NCHELP content for envelope inclusion
VMFCPB4025I Confirming required 7VMCPR20 files are available
VMFCPB4024I Creating file: 7VMCPR20 SERVLINK C
VMFCPB2936I VMFPLCD command processing is in progress, which might require
            several seconds (or minutes) to complete
VMFCPB4003I File 7VMCPR20 SERVLINK C created
VMFCPB4036I Package build for PRODID 7VMCPR20 completed successfully
VMFCPB4027I Initiating package build for product ID 6VMLEN20 (Package 3 of
            3)
VMFCPB4026I Identifying DELTA content for envelope inclusion
VMFCPB4024I Creating file: CSMLEN01 ENVELOPE C
VMFCPB4003I File CSMLEN01 ENVELOPE C created
VMFCPB4024I Creating file: CSMLEN02 ENVELOPE C
VMFCPB4003I File CSMLEN02 ENVELOPE C created
VMFCPB4024I Creating file: CSMLEN03 ENVELOPE C
VMFCPB4003I File CSMLEN03 ENVELOPE C created
VMFCPB4026I Identifying SAMPLEP content for envelope inclusion
VMFCPB4026I Identifying BUILDP content for envelope inclusion
VMFCPB4026I Identifying BUILDL content for envelope inclusion
VMFCPB4026I Identifying NCHELP content for envelope inclusion
VMFCPB4026I Identifying LOCALMOD content for envelope inclusion
VMFCPB4024I Creating file: CSMLEN04 ENVELOPE C
VMFCPB4003I File CSMLEN04 ENVELOPE C created
VMFCPB4025I Confirming required 6VMLEN20 files are available
VMFCPB4024I Creating file: 6VMLEN20 SERVLINK C
VMFCPB2936I VMFPLCD command processing is in progress, which might require
            several seconds (or minutes) to complete
VMFCPB4003I File 6VMLEN20 SERVLINK C created
```

```
VMFCPB4036I Package build for PRODID 6VMLEN20 completed successfully
VMFCPB4133I Performing post-build package processing
VMFCPB4024I Creating file: C5756289 MANIFEST C
VMFCPB4003I File C5756289 MANIFEST C created
VMFCPB2760I CSMPKGBD processing completed successfully (RC=0)
VMFCSV2760I CSMSRVMT EXEC processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

We saw that `ENVELOPE` and `MANIFEST` files were created for the components in the package, and these files were combined with the service content to create SERVLINK files. We were then ready to send the package to our managed system, as described next.

9.3.5 Sending the service package to the managed systems

The **`SERVMGR SRVLVL SEND`** command transmits the service package to the managed system or systems that are serviced.

> **Note:** As we saw earlier when we added the managed system to VMCSM, the MAINT720 ID is logged on by the CSMSERVE user to perform commands. The MAINT720 ID must not be logged on before running service from the principal system. Also, do not attempt to log on to MAINT720 while VMCSM actions are performed.

Example 9-21 shows the result of our **`SERVMGR`** command to send the RSU1 package to RDBKZVMC.

Example 9-21 SERVMGR SRVLVLSEND command output

```
servmgr srvlvl send 720 rsu1 rdbkzvmc
VMFCMG2195I SERVMGR SRVLVL SEND 720 RSU1 RDBKZVMC
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSV2760I CSMSRVMT EXEC processing started
VMFCSV2195I CSMSRVMT SRVLVL SEND 720 RSU1 RDBKZVMC
VMFCTR2760I CSMTRNSP processing started
VMFCTR2195I
VMFCTR3004I Accessing required minidisks/directories...
VMFCTR4080I SEND processing started for system RDBKZVMC
VMFCTR4081I Performing service level check for system RDBKZVMC
VMFCSQ2760I CSMQUERY processing started
VMFCTL4017R Enter the login user ID to be used for RDBKZVMC host system:
            9.76.61.239
csmwork
VMFCTL4018R Enter the login password for user 'csmwork' on RDBKZVMC host
            system: 9.76.61.239
< < password > >
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCTR4081I Performing SFS space allocation check for system RDBKZVMC
VMFCTL4017R Enter the login user ID to be used for RDBKZVMC host system:
            9.76.61.239
csmwork
VMFCTL4018R Enter the login password for user 'csmwork' on RDBKZVMC host
            system: 9.76.61.239
```

```
< < password > >
VMFCTR4033I Filespace 4K block allocation in file pool VMPSFS: is sufficient
VMFCTR4083I Performing file transfers for system RDBKZVMC
VMFCTR4077I Number of files to be transferred: 4
VMFCTR2936I TRANSPORT command processing is in progress, which might require
            several seconds (or minutes) to complete
VMFCTR4078I Transfer results: 4 of 4 files successfully transferred
VMFCTR4084I Initiating installation of package C5756289 for system RDBKZVMC
VMFCTR2936I PKGLOAD command processing is in progress, which might require
            several seconds (or minutes) to complete
VMFCTR4085I SEND processing for system RDBKZVMC is complete
VMFCTR2760I CSMTRNSP processing completed successfully (RC=0)
VMFCSV2760I CSMSRVMT EXEC processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

Reviewing the console of our managed system, we saw that VMCSM logged on the MAINT720 ID several times during the **SEND** action.

If a problem occurs during the package send action, messages are recorded in the $CSMCMG $MSGLOG file. After any problems are rectified, the **SERVMGR SERVLVL SEND** command is run again with the **(RESTART** option.

9.3.6 Putting the service into production

The **SERVMGR SYSTEM PUT2PROD** command is used to take the successfully sent service package and put the service into production. We ran this command on our system and the result is shown in Example 9-22.

Example 9-22 SERVMGR SYSTEM PUT2PROD

```
servmgr system put2prod 720 rdbkzvmc
VMFCMG2195I SERVMGR SYSTEM PUT2PROD 720 RDBKZVMC
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSY2760I CSMSYSMT processing started
VMFCSY2195I CSMSYSMT CMG
VMFCSY4063I SYSTEM PUT2PROD processing started for system RDBKZVMC
VMFCSY4064R Service level RSU1 has a build state of TEST. Enter (1) to put
            the service level into production on system RDBKZVMC or (0) to
            quit
1
VMFCTL4017R Enter the login user ID to be used for RDBKZVMC host system:
            9.76.61.239
csmwork
VMFCTL4018R Enter the login password for user 'csmwork' on RDBKZVMC host
            system: 9.76.61.239
< < password > >
VMFCSY4063I SYSTEM PUT2PROD processing completed successfully for system
            RDBKZVMC
VMFCSY2760I CSMSYSMT processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

Again on the console of the managed system, we saw MAINT720 being logged on. This time, we also saw log on messages for the BLDSEG and BLDCMS IDs, which showed that the **PUT2PROD** command was run by MAINT720.

We checked the status of the service from VMCSM on our principal system and from MAINT720 on the managed system. First, we ran the **SERVMGR SYSTEM QUERY** command against RDBKZVMC to check the VMCSM system status table, as shown in Example 9-23 (we ran the command twice, the second time with the **DETAILS** option to see what other information was now present).

Example 9-23 SERVMGR SYSTEM QUERY after a successful PUT2PROD

```
servmgr system query 720 rdbkzvmc
VMFCMG2195I SERVMGR SYSTEM QUERY 720 RDBKZVMC
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4202I System Information Details for System: RDBKZVMC
VMFCSQ4202I ESM Enablement: NO
VMFCSQ4202I Current Service Level: RSU1
VMFCSQ4202I Pending Service Level: <None>
VMFCSQ4202I Pending Service Level Status: <None>
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
servmgr system query 720 rdbkzvmc (details
VMFCMG2195I SERVMGR SYSTEM QUERY 720 RDBKZVMC (DETAILS
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SYSTEM Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4202I System Information Details for System: RDBKZVMC
VMFCSQ4202I VRM: 720
VMFCSQ4202I Communication Protocol: FTP
VMFCSQ4202I Host Name: 9.76.61.239
VMFCSQ4202I FTP Port: 4535
VMFCSQ4202I FTP Command Operands: <None>
VMFCSQ4202I FTP Data File: <None>
VMFCSQ4202I ESM Enablement: NO
VMFCSQ4202I Current Service Level: RSU1
VMFCSQ4202I Pending Service Level: <None>
VMFCSQ4202I Pending Service Level Status: <None>
VMFCSQ4203I Service Status Information for system: RDBKZVMC
VMFCSQ4203I
VMFCSQ4203I Component        Current From
VMFCSQ4203I Name:            Service Table:
VMFCSQ4203I ---------------- ----------------
VMFCSQ4203I AVSSFS           BASE
VMFCSQ4203I CMSSFS           BASE
VMFCSQ4203I CPSFS            RSU1
```

```
VMFCSQ4203I DIRMSFS          BASE
VMFCSQ4203I DVSFS            BASE
VMFCSQ4203I GCSSFS           BASE
VMFCSQ4203I ICKDSFSFS        BASE
VMFCSQ4203I LESFS            RSU1
VMFCSQ4203I PERFTKSFS        BASE
VMFCSQ4203I RACFSFS          BASE
VMFCSQ4203I REXXSFS          BASE
VMFCSQ4203I RSCSSFS          BASE
VMFCSQ4203I TCPIPSFS         BASE
VMFCSQ4203I TSAFSFS          BASE
VMFCSQ4203I VMHCDSFS         BASE
VMFCSQ4203I VMSESSFS         RSU1
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

We saw that the Current Service Level for the managed system is now RSU1. We also tried a display of the service level by using **SERVMGR SRVLVL QUERY**. The result of this command is shown in Example 9-24.

Example 9-24 SERVMGR Servile QUERY command output

```
servmgr srvlvl query 720 rsu1 (systems
VMFCMG2195I SERVMGR SRVLVL QUERY 720 RSU1 (SYSTEMS
VMFCMG2760I SERVMGR processing started
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFCSQ2760I CSMQUERY processing started
VMFCSQ4205I ----------------------------------------
VMFCSQ4205I CSM SRVLVL Query for z/VM Version 7.2.0
VMFCSQ4205I ----------------------------------------
VMFCSQ4200I Service Level: RSU1
VMFCSQ4200I Root Directory: VMPSFS:CSM720.
VMFCSQ4200I Service Level added on: 10/11/20
VMFCSQ4200I Based on Service Level: BASE
VMFCSQ4200I Service Level last modified on: 10/15/20
VMFCSQ4200I ESM Enablement: NO
VMFCSQ4200I Service Level build state: TEST
VMFCSQ4200I Service Level update lock: No
VMFCSQ4200I Description:
VMFCSQ4200I  Service level: RSU1    Created by: MAINTCSM 10/11/20 23:02:01
VMFCSQ4200I  Based on: BASE
VMFCSQ4200I  Description: RSU LEVEL SHIPPED WITH PRODUCT
VMFCSQ4206I Full service level description available in SERVLVL DESCRIPT
            file in directory: VMPSFS:CSM720.RSU1
VMFCSQ4200I
VMFCSQ4201I Applicable Systems:
VMFCSQ4201I Current: RDBKZVMC
VMFCSQ4201I Pending: <None>
VMFCSQ4200I
VMFCSQ2760I CSMQUERY processing completed successfully (RC=0)
VMFCMG2760I SERVMGR processing completed successfully (RC=0)
Ready;
```

In the Applicable Systems section of the display, we can see that RDBKZVMC is listed as `Current`.

Example 9-25 shows the result of running a **SERVICE ALL STATUS** command by using MAINT720 on the managed system.

Example 9-25 Output of SERVICE ALL STATUS on managed system

```
service all status
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFUTL2767I Reading VMSESE PROFILE B for additional options
VMFSRV2195I SERVICE ALL STATUS
VMFSRV2760I SERVICE processing started
VMFSRV1225I VMSES (7VMSES20%VMSES) status:
VMFSRV1225I    Service Level     RSU1
VMFSRV1225I    Production Level  RDBKZVMC.RSU1
VMFSRV1225I CP (7VMCPR20%CP) status:
VMFSRV1225I    Service Level     RSU1
VMFSRV1225I    Production Level  RDBKZVMC.RSU1
VMFSRV1225I DV (7VMDVF20%DV) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I REXX (7VMREX20%REXX) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I AVS (7VMAVS20%AVS) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I GCS (7VMGCS20%GCS) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I TSAF (7VMTSA20%TSAF) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I CMS (7VMCMS20%CMS) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I TCPIP (7VMTCP20%TCPIP) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I RSCS (7VMRSC20%RSCS) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I DIRM (7VMDIR20%DIRM) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I RACF (7VMRAC20%RACF) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I PERFTK (7VMPTK20%PERFTK) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I VMHCD (7VMHCD20%VMHCD) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV1225I LE (6VMLEN20%LE) status:
```

```
VMFSRV1225I    Service Level     RSU1
VMFSRV1225I    Production Level  RDBKZVMC.RSU1
VMFSRV1225I ICKDSF (5684042J%ICKDSF) status:
VMFSRV1225I    Service Level     BASE
VMFSRV1225I    Production Level  RDBKZVMC.BASE
VMFSRV2760I SERVICE processing completed successfully (RC=0)
Ready;
```

We can see the service and production level designations that correspond to the levels in our service package, and matches the DETAIL information that we saw in the **SERVMGR SYSTEM QUERY (DETAIL** command output that is shown in Example 9-23 on page 298.

Note: When service on a system is managed by VMCSM, the **SERVICE** command is effectively unavailable for interactive use. Only the **SERVICE...STATUS** subcommand is available.

Part 3

System management

In this part, we discuss system management. This part include the following chapters:

10

DirMaint, RACF-connector, and SMAPI

This chapter describes how to enable and configure the DirMaint files to assist you in z/VM user IDs and resource management, configuring the RACF connector for DirMaint to work together with RACF when adding user IDs on z/VM and, configuring DirMaint to use the z/VM Systems Management Application Programming Interface (SMAPI).

If you want to turn on SMAPI, which is required by specific systems management solutions (such as IBM Infrastructure Cloud Center [ICIC] or IBM Wave), you must also have a Directory Maintenance product that was configured as a prerequisite and a RACF connector. DirMaint is described here, but other vendor products are available, such as CA VM:Secure.

Many organizations' security policies require an External Security Manager (ESM) to be implemented and configured. RACF is described here.

This chapter includes the following topics:

10.1 IBM Directory Maintenance Facility

> **Note:** IBM Directory Maintenance Facility (DirMaint) is an optional, priced feature. Before you begin this section, verify that you purchased a license for IBM DirMaint. If you did not, contact your IBM marketing representative for information about how to obtain a license.

Previous versions of this book described manually managing the user directory. For this version, the authors chose not to cover the manual method for the following reasons:

- Manually administering the user directory is needlessly complex, cumbersome, time-consuming, and error prone. Your time is better spent boarding new workloads quickly and optimizing performance versus counting disk cylinders manually and performing data entry.
- The user directory is a fundamentally critical part of z/VM; a corrupted or invalid online user directory can be disastrous. For example, if you inadvertently overlap minidisk definitions, it can cause serious and permanent data loss.
- If your z/VM system has more than a few virtual machines or belongs to a z/VM SSI cluster, it is illogical to attempt manual user management when automation exists.
- For z/VM SSI cluster works correctly, and relocation behaves as designed, the user directories on all members of an SSI cluster must always remain synchronized.

10.1.1 DirMaint features

One of the major advantages of z/VM has always been its ability to provide each user with an individual working environment, a virtual machine (userids). The virtual machine simulates a dedicated, real machine, including processor functions, memory, and input/output resources. Various operating systems and applications can run in a virtual machine.

However, managing many guest operating systems (virtual images) requires a thorough understanding of VM concepts and the knowledge and skill to run a complex set of commands.

When you activate DirMaint, you give control over the z/VM user directory to the DIRMAINT service virtual machine. After that, 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 change the user directory.

DirMaint provides the following features:

- Error checking ensures that only valid changes are made to the user directory.
- Continuous synchronization of changes occurs to all member nodes in an SSI cluster.
- Change authorization permits only authorized personnel to make changes.
- Increased efficiency and productivity are possible through prototypes (and IBM FlashCopy, if available).
- Automatic disk management handles the management of minidisk extents.
- Control of all user-initiated transactions occurs through passwords.
- Logging transactions tracks changes, satisfies governance, and assists with auditing.

10.1.2 DirMaint structure

DirMaint has some userids that are part of product:

- DIRMAINT: When having SSI installed, there should be only one DIRMAINT for each SSI member. which controls the USER DIRECT of z/VM.
- DIRMSAT: When having SSI installed, there should be one DIRMSAT for each SSI member. The first one is DIRMSAT, the other as DIRMSAT2, and so on.
- DATAMOVE: When having SSI installed, there should be at least one DATAMOVE for each SSI member. The first one is DATAMOVE, the other as DATAMOV2, and so on.

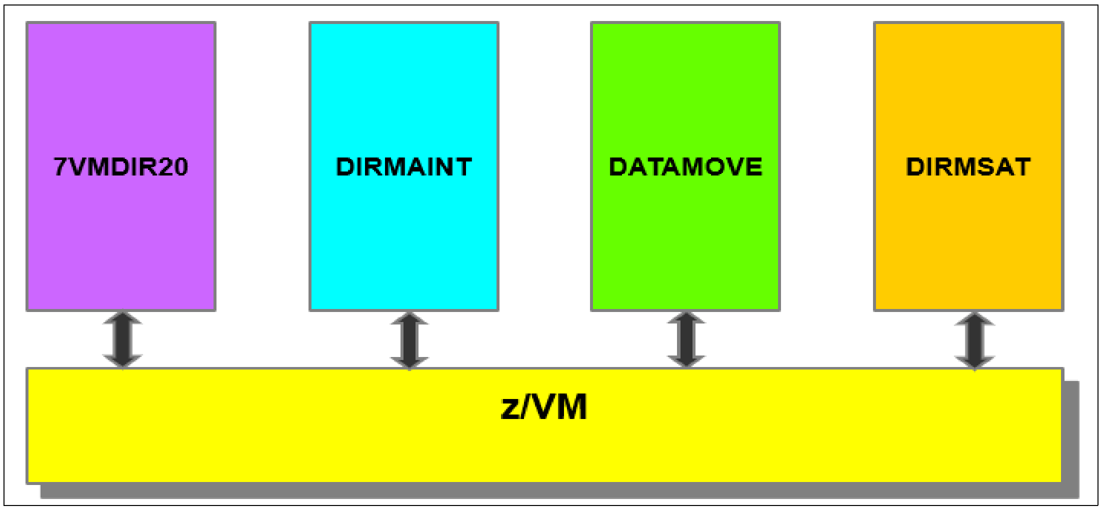


Figure 10-1 Dirmaint Service user IDs

To customize DirMaint, the following configuration files are available:

- CONFIG: DirMaint standard configuration that is included with the product. Do not change this file. If you must customize your installation, create another CONFIG file by using CONFIG plus A - Z, 1 - 9 to form the new CONFIG name, as shown in the following examples:
  - CONFIGn (from A - Z, 1 - 9): Customizable according to your needs; for example, CONFIGAA, CONFIGB, and so on.
  - CONFIGRC: Configuration when having RACF installed.
  - CONFIGSS: Configuration when z/VM SSI is used, it describes the IBM Satellite™ and Datamove user ID s that run in each SSI member.
  - CONFIGSM: Configuration when need to work with SMAPI.
- EXTENT CONTROL: Where you include the DASDs and the name of groups pool.
- AUTHFOR CONTROL: Where you list all of the user IDs that are authorized to issue commands to DIRMAINT.

10.1.3 Finding DirMaint

In this section, we describe how to find the following codes that you need to install and configure DirMaint:

- Dirmaint code

  On the 7VMDIR20 user ID that refers to DirMaint 7.2:

  - MDisks 491/492: Production/test code
  - MDisks 11f/41f: Production/test interface code, all CONFIG file are here.

- Directory files

  On the DIRMAINT user ID:

  - 1df MDisk: Primary source directory files, and other CONTROL files, such as EXTENT, AUTHFOR, and so on.
  - 1db MDisk: Primary location of USER Backup file.

- Directory utilities
- On the PMAINT user ID: 551 MDisk (DIRECTXA, DIRMAP, and DISKMAP utilities).

For more information, see *Program Directory for Directory Maintenance Facility for z/VM function level 720 Program Number 5741-A09 for Use with z/VM version 7 release 2*, GI13-4362.

10.1.4 Enabling DirMaint

You must enable DirMaint on the first member of the SSI cluster only. Other SSI members have DirMaint satellite servers that send user directory update requests to the member where DirMaint is running.

DirMaint is included in a disabled state with z/VM. To enable it, complete the following steps:

1. Log on as **MAINT720** on member 1 of the z/VM SSI cluster.

   Important: You must use the `MAINT720` ID to perform these steps. Do not use `MAINT` or any other ID.

2. Verify that the MAINT `51D` minidisk is accessed as file mode D and is read/write (R/W):

```
QUERY ACCESSED
Mode  Stat     Files  Vdev  Label/Directory
A      R/W        71  191   MNT191
B      R/W       134  5E6   MNT5E6
C      R/O        19  2CC   MNT2CC
D      R/W       299  51D   MNT51D
E      R/W        12  551   PMT551
S      R/O       698  190   MNT190
Y/S    R/O      1123  19E   MNT19E
```

   If `MNT51D` is not shown at all, or is Read Only (R/O), use **VMLINK** to correct the situation and then, reissue the **QUERY ACCESSED** command to verify results.

```
VMLINK MAINT 51D < 51D D MR >
DMSVML2060I MAINT 51D linked MR as 051D file mode D
```

> **Important:** If VMLINK returns **HCPLNM103E DASD 051D forced R/O**, another user has the link as R/W and must change its access to R/O. Do not use MW mode for any MDisk at all, you can destroy the MDisk content.

3. Enable DirMaint through the VMSES/E **SERVICE** command. Ensure that the message VMFSRV2760I is displayed:

```
===> service dirm enable
...                      // several windows full of text will quickly go by
VMFSRV1233I The following products have been serviced.
VMFSRV1233I DIRM
VMFSRV2760I SERVICE processing completed successfully.
```

4. Put DirMaint into production with the **PUT2PROD** command. Ensure that the message VMFP2P2760I is displayed:

```
===> put2prod dirm
VMFP2P2760I PUT2PROD processing started
several messages ...
VMFP2P1233I The following products have been put into production. Recycle the
            appropriate servers.
VMFP2P1233I DIRM
several messages ...
VMFP2P2760I PUT2PROD processing completed successfully (RC=0).
```

5. Optional: Review the changed SYSTEM CONFIG file. The SERVICE and PUT2PROD steps modified data that is near the end of your SYSTEM CONFIG file, regarding the enablement of DirMaint.

   If you want to see the changes, link to the PMAINT CF0 disk and use the **type** command to output the contents of the SYSTEM CONFIG file to observe these lines at the end of the file:

```
===> vmlink pmaint cf0 (filelist
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
===> xedit system config z
===> bot
===> u10
PRODUCT PRODID 7VMDIR20 STATE ENABLED DESCRIPTION '09/23/20.12:18:34.MAINT720
Install/service DirMaint using minidisk'
```

> **Tip:** You can also perform this step in only one command instead of three commands. Note that the parenthesis part of the command and must be included:
>
> `===> vmlink pmaint cf0 (invoke type system config z`

6. Log off from MAINT720 on z/VM system. If you are in z/VM SSI cluster, you need to log on member1.

7. Log on to the next member in the SSI cluster as **MAINT720** and put DirMaint into production on that node by issuing the **PUT2PROD DIRM** command. Repeat this action for each SSI member in the cluster.

DirMaint is now enabled across all members of the SSI cluster.

10.2 Tailoring DirMaint

This section describes the DirMaint tailoring in an SSI cluster or on stand-alone systems. Consider the following points:

- When mentioning member 1 of the cluster, you can run the same instruction on stand-alone systems.
- When describing actions to other members, we are referring only to the members in the SSI cluster.

10.2.1 Changing default passwords

This section describes changing the default password on USER DIRECT file, regardless of whether you plan to use DirMaint.

To take the first major step toward correctly securing your new z/VM system, complete the following steps to change the default passwords for Service Virtual Machines (SVMs):

1. Log on as **MAINT720** on the first member of the SSI cluster.
2. Verify that the `MAINT 2CC` minidisk is accessed as file mode C and is read/write (R/W):

```
===> query accessed
Mode  Stat     Files  Vdev  Label/Directory
A      R/W        71  191   MNT191
B      R/W       134  5E6   MNT5E6
C      R/W   19  2CC   MNT2CC
D      R/W        299 51D MNT51D
E      R/W        12  551   PMT551
S      R/O       698  190   MNT190
Y/S    R/O      1123  19E   MNT19E
```

   If you find that `MNT2CC` is not shown, or is Read Only (R/O), use `VMLINK` to correct the situation and then reissue `QUERY ACCESSED` to verify the results:

```
===> VMLINK MAINT 2CC < 2CC C MR >
DMSVML2060I MAINT 2CC linked MR as 02CC file mode C
```

3. Open the z/VM user directory for editing:

```
===> xedit user direct C
```

4. Change the passwords of `7VMDIR20`, `DIRMAINT`, `DIRMSAT`, `DIRMSAT`*x* (where *x* is `2`, `3`, or `4`, depending on the number of SSI member nodes), `DATAMOVE`, and `DATAMOVx` from their current value (typically `AUTOONLY`) to an eight character value of your choice. These IDs are powerful, so choose non-trivial values.

```
===> /user 7vmdir20
USER 7vmdir20 <NEWPASWD>  16M   64M EG

===> top
===> /user dirmaint
...
```

5. Change the passwords of all service machines that might use the default. The default password for your system is contained in the information that was provided to you at the time of purchase. In this example, the default password is `MDRKI9OP`. This command is case-sensitive:

```
===> C/MDRKI9OP/NEWPASWD/**
DMSXCG517I ## occurrence(s) changed on ## line(s)
```

6. Run the **DIRECTXA** command as `MAINT720` on all members to bring the changes online:

```
===> DIRECTXA USER DIRECT
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 7 RELEASE 2.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 56 disk pages
```

10.2.2 Configuring DirMaint

Complete the following steps to configure DirMaint:

1. Log on as **7vmdir20** on the first member of the SSI cluster by using the new password that you set in section 10.2.1, “Changing default passwords”.
2. Access the `492` disk as `K` to get access to the **DIR2PROD EXEC** by using the following command:

```
===> access 492 K
```

3. Use the following **DIR2PROD EXEC** command to access the necessary minidisks:

```
===> dir2prod access_new 7vmdir20 dirm
DMSACP726I 492 K released
DIR2PROD: Normal Termination.
```

4. Three new minidisks were accessed as `J`, `K`, and `L`:

```
===> query accessed
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         3  DIR   VMPSFS:7VMDIR20.
B      R/O       140  5E5   MNT5E5
D      R/O       210  51D   MNT51D
J      R/W        14  1DF   DIR1DF
K      R/W       286  492   DRM492
L      R/W        55  41F   DRM41F
S      R/O       693  190   MNT190
Y/S    R/O      1121  19E   MNT19E
```

5. Create the primary DirMaint local customization parameters file, **CONFIGAA DATADVH L**. The `L` disk needs to be `DIRMAINT 41F`, which is the preproduction disk. Add the lines that are shown in Example 10-1. Press Enter after each line. After you add all of the lines, press Enter twice to end INPUT mode, and type `FILE` to save:

```
===> xedit CONFIGAA datadvh L
===> input
   DISK_CLEANUP= YES
   PW_INTERVAL_FOR_SET= 90
   ONLINE= IMMED
   RUNMODE= OPERATIONAL
   RACF_RDEFINE_VMBATCH_DEFAULTS=
   MESSAGE_LOGGING_FILETYPE=           TRANSLOG
   MESSAGE_LOG_RETENTION_PERIOD=       3  (MONTHS)
```

```
PURGE_COMMAND_PROCESSING=            FULL
SHUTDOWN_MESSAGE_FAILURE=            LOGOFF
DATAMOVE_MACHINE=                    DATAMOVE * *
```

```
===> file
```

- DISK_CLEANUP= YES ensures privacy by cleaning up residual data, but also means that changes will take longer while DirMaint reformats any abandoned minidisk extents.
- PW_INTERVAL_FOR_SET= 90 sets a 90-day password change interval. If you plan to use an ESM, such as RACF, you need to omit this line entirely because the ESM will handle this interval.
- ONLINE= IMMED line sets your changes to be made immediately.
- RUNMODE= OPERATIONAL indicates that directory changes need to be made. This run mode can be set to TESTING and the changes will not be performed yet. If you use testing mode, ensure that you remember to come back and change to operational mode when your testing is complete.
- The RACF_RDEFINE_VMBATCH_DEFAULTS= line does not create a VMBATCH-specific resource entry. Otherwise, DIRMAINT creates a VMBATCH resource for this user ID with this line as a default. The VMBATCH generic resource class is configured in , "This output shows that SMAPI is running, LNXADMIN is correctly authorized to call SMAPI, and the Linux interface **smaclient** is working." on page 340. If you are not installing RACF, you can omit this line.

Example 10-1 Example CONFIGAA DATADVH file contents

```
DISK_CLEANUP= YES
PW_INTERVAL_FOR_SET= 90
ONLINE= IMMED
RUNMODE= OPERATIONAL
RACF_RDEFINE_VMBATCH_DEFAULTS=
MESSAGE_LOGGING_FILETYPE=            TRANSLOG
MESSAGE_LOG_RETENTION_PERIOD=        3  (MONTHS)
PURGE_COMMAND_PROCESSING=            FULL
SHUTDOWN_MESSAGE_FAILURE=            LOGOFF
DATAMOVE_MACHINE=                    DATAMOVE * *
```

TIP: Some definitions are required for compliance reasons and it can vary upon your security policies (for example, the log retention period and disk cleanup settings). You might also consider other options, such as altering the output line length, as described in 15.5, "System modifications for wide-screen terminals" on page 449

6. Copy CONFIGAA DATADVH to the 11F minidisk:

```
===> ACC 41F L
===> ACCESS 11F F
===> COPY CONFIGAA DATADVH L = = F (OLDDATE
```

10.2.3 Working with DirMaint AUTHFOR file

The AUTHFOR CONTROL file specifies user IDs that are allowed to perform authorized DirMaint tasks. Only grant authorization to the user IDs that have administration roles on z/VM. You can add user IDs when needed.

Create the `AUTHFOR CONTROL` file on the `J` disk (`DIRMAINT 1DF`). Add 8 lines to accommodate the entries that are shown in Figure 10-2 for all IDs that are required to perform DirMaint tasks.

```
===> xedit authfor control j
===> a 8
```

```
ALL MAINT720 * 140A ADGHOPS
ALL MAINT720 * 150A ADGHOPS
ALL MAINT    * 140A ADGHOPS
ALL MAINT    * 150A ADGHOPS
ALL LNXADMIN * 140A ADGHOPS
ALL LNXADMIN * 150A ADGHOPS
ALL LNXMAINT * 140A ADGHOPS
ALL LNXMAINT * 150A ADGHOPS
```

Figure 10-2 List of entries to add into the AUTHFOR CONTROL file

A command level of `140A` allows the authorized user to enter commands by using DirMaint Release 4 compatibility syntax. A command level of `150A` allows the authorized user to enter commands by using the DirMaint Release 5 full-function syntax. It is recommended to give access to include records for both 140A and 150A command levels for each target ID/authorized user pair. Entries that are added to this file do not need to necessarily exist in the User Directory yet, so do not worry that undefined entries are being added.

If you are working with IBM ICIC (IBM Cloud Infrastructure Center) or IBM Wave (to graphically manage your system), you can add the user IDs of ICIC, WAVE, and SMAPI.

Many of the DirMaint configuration files are now created. The next important file is the `EXTENT CONTROL` file. which is discussed next.

10.2.4 Customizing the EXTENT CONTROL file

The `EXTENT CONTROL` file defines disks (volumes) to DirMaint for minidisk allocation. It also contains system and device default values that are used during allocation operations.

Two main sections must be populated:

Regions Defines the actual disks and their sizes to DirMaint. The `AUTOR` keyword can be used in user directory entries to take space from the regions. It is recommended that region name and volume label are always identical.

Groups Defines pools of disks so the `AUTOG` keyword can be used to take space from the pools, not from specific disks.

To configure the EXTENT CONTROL file, complete the following steps:

1. Issue the **QUERY DASD** command to see the disks that are attached to SYSTEM. Disregard the CP-owned DASD and the common volumes. Write down the output or copy and paste it out of your 3270 emulator into a new text document on your workstation because you will need to refer to it while you perform the next steps:

```
===> query dasd
DASD  953E CP OWNED  RS3CM1   23
DASD  95BE CP SYSTEM RS3RL1   28
DASD  963E CP OWNED  RDGRES   105
DASD  96BE CP OWNED  RDGS01   1
DASD  973E CP OWNED  RDGP01   0
DASD  983E CP OWNED  RDHS01   0
```

2. Make a copy of the original EXTENT CONTROL file:

```
===> copy extent control j = contorig = (olddate
```

3. Add the DASD that is attached to SYSTEM in the :REGIONS. section (assuming that these volumes will be available for minidisk creation or several of the default system minidisks are present). The convention that is used in this example is that the RegionID, field 1, is set to the VolSer, field 2. Fields 3 and 4 set the cylinder range to all cylinders except cylinder 0, and the Dev-Type, the last field, informs DirMaint of the size of the disk. Each region name is also added to one or more GROUPs.

   If you are not sure of the device type, use the **QUERY DASD DETAILS <rdev>** command from MAINT or MAINT720 user ID that has the CP class to issue this command. Example 10-2 is an example of a pipe command to get the size information. You can put the DASD addresses individually or issue the command by using a contiguous DASD range.

   Example 10-2 Example of a pipe to get the size information of each disk

```
pipe cp q da details 9432 94b2 9532 953e 95b2 9632 96b2 | locate /CYLS =/| cons
9432  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBCM1, CYLS = 10017
94B2  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBRL1, CYLS = 10017
9532  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBRL2, CYLS = 10017
953E  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = RS3CM1, CYLS = 10017
95B2  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBRES, CYLS = 10017
9632  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBS01, CYLS = 10017
96B2  CUTYPE = 2107-E8, DEVTYPE = 3390-0E, VOLSER = VMBP01, CYLS = 10017
```

4. DirMaint provides the capability to clone a SUBCONFIG entry by using an existing SUBCONFIG entry. By using the **:SSI_VOLUMES** section, you can define the DASD volumes that DirMaint will use when it allocates the new minidisks that are associated with the new cloned SUBCONFIG entry. Entries within the **:SSI_VOLUMES** section define the DASD volume that corresponds to the user-defined set of volumes across each member of the SSI cluster. If your system is z/VM stand-alone, ignore this section.

   Example 10-3 on page 315 shows the commands that are used to:

   – List all SFS directories and minidisks currently accessed.

   – Open the EXTENT CONTROL file in edit mode, allowing you to make modifications where necessary. For example, if you need to add DASD volumes that correspond to the user-defined set of volumes across each member of the SSI cluster.

> **Important:** As you enter your Dev-Type values, you *must* use two digits for the Type. The 3390 model 3 regions must be entered as 3390-03, 3390 model 9 regions must be entered as 3390-09.

Example 10-3 Listing the volumes and opening the EXTENT CONTROL file for editing

```
===> query accessed
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         3  DIR   VMPSFS:7VMDIR20.
B      R/O       140  5E5   MNT5E5
D      R/O       210  51D   MNT51D
J      R/W        14  1DF   DIR1DF
K      R/W       286  492   DRM492
L      R/W        55  41F   DRM41F
S      R/O       693  190   MNT190
Y/S    R/O      1121  19E   MNT19E

===> xedit extent control j
* *********************************************************************
...
Purpose:   Default Extent Control file.
...
* *********************************************************************
:REGIONS.
*RegionId  VolSer    RegStart      RegEnd  Dev-Type  Comments
VMBCM1    VMBCM1    0001          END     3390-09
 VMBRL1    VMBRL1    0001          END     3390-09
 VMBRL2    VMBRL2    0001          END     3390-09
 RS3CM1    RS3CM1    0001          END     3390-09
 VMBRES    VMBRES    0001          END     3390-09
 VMBS01    VMBS01    0001          END     3390-09
 VMBP01    VMBP01    0001          END     3390-09
:END.
:GROUPS.
*GroupName RegionList
* SYSTEM is for z/VM System Volumes
SYSTEM   VMBCM1 VMBRL1 VMBRL2 RS3CM1 VMBRES VMBS01 VMBP01
* USRWORK group is for z/VM Work Volumes on all members (in our lab environment
are G and H z/VM members):
USRWORK RDGUS1 RDGUS2 RDHUS1 RDHUS2
* LNXADM1 group is for full-pack minidisks used by LNXADM-1
LNXADM1  VM1567
* LNXADM2 group is for full-pack minidisks used by LNXADM-2
LNXADM2  VM1569
* POOL1 is a group for Linux virtual machines
POOL1    VM156A VM156B VM156C VM156D
POOL1    VM156E VM156F
* POOL2 is a group for Kiwi
POOL2             VM1222
:END.
:SSI_VOLUMES.
* Added during Installation, Do not remove.
*VolumeFamily     Member   VolSer
IBM_RES           RDBKZVMG RDGRES
IBM_WORK1         RDBKZVMG RDGUS1
IBM_RES           RDBKZVMH RDGRES
IBM_WORK1         RDBKZVMH RDHUS1
:END.
:EXCLUDE.
* ENTRY_NAME ADDRESS
```

```
MAINT*       012*
MAINT*       013*
PMAINT       013*
PMAINT       014*
SYSDUMP1     012*
SYSDMP*      012*
:END.
:AUTOBLOCK.
* IBM supplied defaults are contained in the AUTOBLK DATADVH file.
* The following are customer overrides and supplements.
*
*DASDType BlockSize Blocks/Unit Alloc_Unit Architecture
:END.
:DEFAULTS.
* IBM supplied defaults are contained in the DEFAULTS DATADVH file.
* The following are customer overrides and supplements.
*
*DASDType Max-Size
   3390-03  3339
   3390-09  10017
   3390-27  30051
   3390-54  60102
:END.
```

5. Update the DirMaint configuration:

```
===> dir2prod update_files 7vmdir20 dirm
DIR2PROD: Matched   CONFIG   SAMPDVH  F  with CONFIG   SDV11501 G2
DIR2PROD: Replacing CONFIG   SAMPDVH  F  with CONFIG   SDV11501 G2
DIR2PROD: Matched   CONFIG   DATADVH  F  with CONFIG   SDV11501 G2
...
DIR2PROD: Matched   LINDFLT  DIRECT   J  with LINDFLT  SAMPDVH  H2
DIR2PROD: Leaving   LINDFLT  DIRECT   J  unchanged.
DIR2PROD: Normal Termination.
```

6. Copy the DirMaint configuration:

```
===> dir2prod prod 7vmdir20 dirm
DIR2PROD: Copy of 492 disk to 491 disk has started.
  Command: VMFCOPY * EXEC K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * REXX K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * XEDIT K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATADVH K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATAADVH K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATAUDVH K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * MODULE K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY DVHPROFA * K = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
DIR2PROD: Copy of 492 disk to 491 disk has completed.
DIR2PROD: Copy of 41F disk to 11F disk has started.
  Command: VMFCOPY * EXEC L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * XEDIT L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATADVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATAADVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATAKDVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * DATAUDVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * MSGADVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * MSGKDVH L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
```

```
  Command: VMFCOPY * MODULE L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * NEWMAIL L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
  Command: VMFCOPY * REXX L = = E (PRODID 7VMDIR20%DIRM OLDDATE REPLACE
DIR2PROD: Copy of 41F disk to 11F disk has completed.
DIR2PROD: Normal Termination.
```

7. Log off from `7vmdir20`.

The `EXTENT CONTROL` file, which is read when DirMaint starts, is now configured.

> **Note:** For more information about all DirMaint files, see Appendix H of *z/VM Version 7 Release 2 Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283.

10.2.5 Copy User Direct to be initialized by DirMaint

Complete the following steps to put the USER DIRECT file under control of DirMaint:

1. Log on to the user ID MAINT720.
2. Issue the command **`q disk`** or **`q accessed`** to check whether 2CC MDisk is accessed as letter C, as shown in Example 10-4.

Example 10-4 Q DISK command on MAINT720 user ID

```
q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
MNT191 191  A   R/W   175 3390 4096        3        10-01      31490      31500
MNT5E5 5E5  B   R/O    40 3390 4096      140      1547-21       5653       7200
MNT2CC 2CC  C   R/O    10 3390 4096        4        98-05       1702       1800
MNT51D 51D  D   R/O    26 3390 4096      210      1189-25       3491       4680
PMT551 551  E   R/O    40 3390 4096       10       132-02       7068       7200
MNT190 190  S   R/O   207 3390 4096      693     15970-43      21290      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1121     35137-39      54863      90000
Ready; T=0.01/0.01 12:15:47
```

3. If it is not accessed as shown in Example 10-4, access the user directory source file (USER DIRECT) that is on P`MAINT 2CC` MDisk by using the **`VMLINK`** command. The read password is the value that you set all passwords to, or if you did not change them, it is **`READ`**:

   ```
   ===> vmlink pmaint 2CC <2CC C>
   DMSVML2060I MAINT 2CC linked as 0120 file mode C
   ```

   > **Hint:** The VMLINK command links the MDisk with another MDisk address. You must access it with the filemode (in our case, **`C)`**. Therefore, you must issue **`filelist * * c`** to access the content of this MDisk.

4. Link to `DIRMAINT 1df` disk, as shown in Example 10-5.

Example 10-5 Disks accessed

```
vmlink dirmaint 1df (w
DMSVML2060I DIRMAINT 1DF linked as 0120 file mode Z

q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
MNT191 191  A   R/W   200 3390 4096       22        90-01      35910      36000
MNT5E6 5E6  B   R/W    20 3390 4096      140      1547-43       2053       3600
```

```
MNT2CC 2CC  C   R/O    10 3390 4096        4        98-05      1702       1800
MNT51D 51D  D   R/W    26 3390 4096      210      1189-25      3491       4680
PMT551 551  E   R/W    40 3390 4096       12       132-02      7068       7200
MNT190 190  S   R/O   207 3390 4096      693     15970-43     21290      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1121     35137-39     54863      90000
DIR1DF 121  Z   R/W    12 3390 4096       14        97-04      2063       2160
```

5. Copy the `USER DIRECT` file from `MAINT 2CC` (file mode `C`) to `DIRMAINT 1DF` (file mode `Z R/W`) as the file `USER INPUT`, which causes the current user directory to be loaded into DirMaint when it starts for the first time, as shown in Example 10-6.

```
===> copy user direct C = input Z
```

Example 10-6 Copying the USER DIRECT file

```
copy user direct C = input Z

filelist USER * *
MAINT720 FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      USER     INPUT    Z1 F         80       3847         76  9/23/20 18:01:25
      USER     DISKMAP  A1 F        100        327          8  9/15/20 20:22:21
      USER     DIRECT   C1 F         80       3847         76  9/09/20 11:33:46
      USER     DISKMAP  C1 F        100        327          8  9/08/20 17:56:46
```

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 might rename it to `USER DIR_ORIG` on the same `2CC` MDisk.

10.3 Starting DirMaint

To start DirMaint, complete the following steps:

1. Log on as `MAINT` on the first SSI member.
2. Issue the following command, which is two separate commands. The command on the left half of the number sign (#), which is the line-end character, starts `DIRMAINT` with the **`XAUTOLOG`** command and the **`SYNC`** option returns control to `MAINT`. The second command on the right side of the # sets `MAINT` to be the secondary user of `DIRMAINT`. This way, `DIRMAINT` does not need to be logged on to, but `MAINT` can see its console output:

```
===> xautolog dirmaint sync # set secuser dirmaint *
AUTO LOGON  ***       DIRMAINT USERS = 13
Ready;
HCPCFX6768I SECUSER of DIRMAINT initiated.
Ready;
......................................................................
DIRMAINT: PRODUCT:
DIRMAINT: IBM Directory Maintenance Facility for z/VM (DirMaint)
......................................................................
DIRMAINT: DIRMAINT ENDVM363. - 2015/04/14; T=0.01/0.01 18:02:15
 DIRMAINT: DVHILZ3510I Starting DVHINITL with directory: USER INPUT E
 DIRMAINT: DVHILZ3510I DVHINITL Parms: BLDMONO NOCRCWARN
...
```

```
DIRMAINT: DVHWAI2140I Waiting for work on 20/09/23 at 18:03:45.
```

> **Note:** Watch for errors. Look for the message that suggests that the DirMaint directory is initialized by using the file `USER INPUT`, which was copied from `USER DIRECT` earlier.

3. Turn off the secondary user setting so `MAINT` will no longer see the `DIRMAINT` console messages:

```
===> set secuser dirmaint off
DIRMAINT: HCPCFX6769I Your SECUSER terminated by MAINT.
HCPCFX6769I SECUSER of DIRMAINT terminated.
```

DirMaint is now running. It read the `USER INPUT`, `CONFIGAAA DATADVH`, `AUTHFOR CONTROL`, and `EXTENT CONTROL` configuration files.

If you want, you can check the DIRMAINT and DIRMSAT# virtual service machines (VSMs) across all member nodes by using the following commands:

```
===> query DIRMAINT at all
===> query DIRMSAT1 at all
```

and so on, if you have more z/VM lpars with DIRMSAT2 and DIRMSAT3.

> **Important:** From this point forward, you must not attempt to directly (manually) edit any copies of USER DIRECT nor attempt to use the DIRECTXA command. After a directory is initialized, direct editing introduces checksum errors, possibly for every entry if default serialization is allowed to occur.

10.3.1 Validating DirMaint

To validate your DirMaint installation, complete the following steps:

1. (Optional) Run the following commands to update the terminal characteristics for `MAINT` and `MAINT720` so that it is easy to distinguish when you are logged on to these highly privileged IDs:

```
===> dirmaint for maint COMMAND ADD 10 SCREEN STAT RED REV
DVHXMT1191I Your COMMAND request has been sent for processing ...
Ready; T=0.01/0.01 18:31:33
 DVHREQ2288I Your COMMAND request for MAINT at * has been accepted.
 DVHBIU3450I The source for directory entry MAINT has been updated.
 DVHREQ2289I Your COMMAND request for MAINT at * has completed; with RC = 0.
===> dirmaint for maint COMMAND ADD 10 SCREEN INAR YEL UND
===> dirmaint for MAINT720 COMMAND ADD 10 SCREEN STAT RED REV
===> dirmaint for MAINT720 COMMAND ADD 10 SCREEN INAR YEL UND
```

2. Run the **`DIRMAINT REVIEW`** command to spool a file, which contains an overview of the directory entry for MAINT, to `MAINT`'s reader. No prompt for a password occurs:

```
===> dirmaint for maint review
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT at ...
Ready;
 DVHREQ2288I Your REVIEW request for MAINT at * has been accepted.
RDR FILE 0009 SENT FROM DIRMAINT PUN WAS 3397 RECS 0117 ...
 DVHREQ2289I Your REVIEW request for MAINT at * has completed; with RC = 0.
```

3. Use the **PEEK** command with the file number that was sent to the reader to view the contents of the file. In this example, the file number is ***0009***. The **(FOR *** option parameter specifies not to truncate during viewing (so you can view all lines):

```
===> peek 0009 (FOR *
IDENTITY MAINT XXXXXXXX 256M 1000M ABCDEFG
DVHRXV3366I The following configurations will be used on SSI nodes.
...
```

> **Tip:** You can enter `rdrlist` at the CMS ready prompt to view all of the files that are currently spooled to the reader. By moving your cursor to any line and pressing PF11, you can invoke `PEEK` for that file.

4. While you are still inside PEEK, when you are finished looking at the review file, issue the command **DISCARD** to exit out of PEEK and then remove the file from the reader:

```
===> DISCARD
File MAINT DIRECT has been discarded
```

5. Query DirMaint for the listing of DASD groups that were defined in EXTENT CONTROL:

```
===> dirmaint DASD QUERY GROUP *
```

6. Query the current STORAGE values that were set for DIRMAINT:

```
===> dirmaint for dirmaint storage ?
...
DVHSTO3207I DIRMAINT currently has a maxstorage value of 256M and a
DVHSTO3207I default storage value of 128M.
DVHREQ2289I Your STORAGE request for DIRMAINT at * has completed; with RC = 0.
```

7. Test for user and device locks, then test the status of the DATAMOVE worker machines:

```
===> dirmaint status locked both
DVHXMT1191I Your STATUS request has been sent for processing to DIRMAINT ...
Ready;
 DVHREQ2288I Your STATUS request for MAINT at * has been accepted.
 DVHSTT3416I There are no User locks currently active.
 DVHSTT3416I There are no device locks currently active.
 DVHREQ2289I Your STATUS request for MAINT at * has completed; with RC = 0.

===> dirmaint status datamove all
DVHXMT1191I Your STATUS request has been sent for processing to DIRMAINT ...
DVHREQ2288I Your STATUS request for MAINT at * has been accepted.
 DVHSTT3418I DATAMOVE RDBKZVMG Sysaffin: * Activity: INACTIVE Pending: 0
 DVHSTT3418I CurUnit:  Autolog Attempts: 0
 DVHSTT3418I DATAMOV2 RDBKZVMH Sysaffin: * Activity: INACTIVE Pending: 0
 DVHSTT3418I CurUnit:  Autolog Attempts: 0
 DVHREQ2289I Your STATUS request for MAINT at * has completed; with RC = 0.
```

These tests show that DirMaint is configured and functioning.

10.4 DirMaint-RACF Connector

> **Note:** Only configure RACF-Connector if you completed the steps to enable RACF as described in 6.7, "Enabling and configuring RACF" on page 155.
>
> If you do not plan to enable RACF, skip this section.

Tailoring your DirMaint system includes implementing security measures against unauthorized access to data, and inadvertent destruction of data.

DirMaint itself provides a level of security through its command set authorizations. These can be tailored to suit the using installation's needs. However, for critical data files, extra security measures must be implemented. This process can be done by using an External Security Manager (ESM), such as Resource Access Control Facility (RACF).

An ESM controls who can have access, and what kind of access they can have to specific data files and disks. If an ESM is implemented at your installation, DirMaint must be given the appropriate access to the disks and files you want it to manage.

DirMaint can call RACF for the following functions:

- User add or change
- Password or passphrase change
- Logonby change
- POSIX parameter change
- Minidisks commands (AMDISK, CMDISK, DMDISK, etc)

You need RACF or another ESM installed before implementing what is described in this chapter. To enable and configure RACF, see 6.7, "Enabling and configuring RACF" on page 155.

If RACF is installed on your system as the ESM, several entries in the CONFIG DATADVH file set defaults for the DirMaint RACF connector support, which provides automatic communication with RACF.

> **Note:** These recommendations are optional and whether you follow them depends on the level of security that your installation requires.

10.4.1 Configuring RACF-Connector

RACF can coexist with the DirMaint product installed. However, to avoid dual maintenance of password processing (and other RACF functions), complete the following steps:

1. Use the DirMaint supplied sample file `CONFIGRC SAMPDVH`. You must copy this file to the 7VMDIR20 11F disk as `CONFIGRC DATADVH`.
2. If RACF administration is centralized, you must give the DIRMAINT user ID the RACF attribute `SPECIAL`. If RACF administration is decentralized, you must give the DIRMAINT user ID RACF `group-SPECIAL` attribute.
3. If you want to record DirMaint activity in RACF SMF records, enable `ESM_LOG_RECORDING_EXIT`. For this change to take effect, run the **`DIRM RLDDATA`** command.
4. You must also authorize the DirMaint service machines DIRMAINT, DATAMOVE, and DIRMSAT to use the RACROUTE interface.

Note: For more information, see Directory Maintenance Facility Tailoring and Administration Guide, SC24-6135 and z/VM: Security Server RACROUTE Macro Reference, SC24-6231.

10.4.2 Adding RACF connector configuration

Note: To create a configuration file or update them on MDisk DIRMAINT 1DF, you must have it as read/write and DIRMAINT is not up and running. To avoid this issue, work on your MDisk A and create the files there. Then, send the file to DirMaint as described in this section.

When DirMaint is not up yet

Create another DirMaint configuration file, `CONFIGRC DATADVH L` to have different configuration file for RACF only. The L disk is on DIRMAINT 41F, which is the pre-production disk. Add the following lines:

```
===> vmlink dirmaint x configrc datadvh l
===> a 10
USE_RACF= YES ALL
RACF_ADDUSER_DEFAULTS=                  UACC(NONE)
RACF_DISK_OWNER_ACCESS=                 ACC(ALTER)
RACF_RDEFINE_VMPOSIX_POSIXOPT.QUERYDB=  UACC(READ)
RACF_RDEFINE_VMPOSIX_POSIXOPT.SETIDS=   UACC(NONE)
RACF_RDEFINE_SURROGAT_DEFAULTS=         UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VMBATCH_DEFAULTS=          UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VMRDR_DEFAULTS=            UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VMMDISK_DEFAULTS=          UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VSWITCH_LAN=               YES | NO
PW_WARN_MODE=                           MANUAL
PW_LOCK_MODE=                           MANUAL
ESM_PASSWORD_AUTHENTICATION_EXIT=       DVHXPA   EXEC
/ESM_LOG_RECORDING_EXIT=                DVHESMLR EXEC
PASSWORD_CHANGE_NOTIFICATION_EXIT = DVHXPN EXEC
USER_CHANGE_NOTIFICATION_EXIT = DVHXUN EXEC
```

```
===> file
```

Your new configuration file was created successfully on DIRMAINT 1DF mdisk.

When DirMaint is up

Create the same DirMaint configuration file, `CONFIGRC DATADVH A`, on your MDisk A. Then, send them to DIRMAINT. This process is done because the MDisk 1DF is accessed as read/write by DirMaint and cannot be linked as read/write at the same time. Add the following lines:

```
===> x configrc datadvh a
===> a 10
(include the same lines listed before)
===> file

To send the file to DirMaint you need:
===> dirm file CONFIGRC DATADVH
To load the file to DirMaint you need:
```

```
===> dirm rldc
```

Your new configuration file was created successfully on DIRMAINT 1DF MDisk.

10.4.3 Verifying that DirMaint and RACF work together

To add virtual machines, you must use DirMaint and RACF commands. Complete the following steps:

1. Log in as MAINT.
2. Create a sample virtual machine prototype that is named LNXSAMPL PROTODIR:

   ```
   ===> x lnxsampl protodir a
   USER LNXSAMPL LNX4VM 256M 2G G
       INCLUDE LNXDFLT
       MDISK 0100 3390 AUTOG 10016 POOL1 MR LNX4VM LNX4VM LNX4VM
       MDISK 0101 3390 AUTOG 10016 POOL1 MR LNX4VM LNX4VM LNX4VM
   ```

   This definition gives each Linux virtual machine 256 MB of initial memory (with up to 2 GB dynamic memory) and two 3390-9 disks or about 14 GB of disk space. The **AUTOG** and **POOL1** keywords instruct DirMaint to automatically choose space from the pool of volumes in the pool that is named POOL1.

3. Register the prototype with DirMaint by using the **DIRM FILE** command:

   ```
   ===> dirm file lnxsampl protodir
   10:08:53 PUN FILE 0069 SENT TO   DIRMAINT RDR AS  0086 RECS 0012 CPY  001 0
   NOHO
   LD NOKEEP
   DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
   DVHXMT1191I at POKDEV62.
    DVHREQ2288I Your FILE request for MAINT at * has been accepted.
    DVHRCV3821I File LNXSAMPL PROTODIR A has been received; RC = 0.
    DVHREQ2289I Your FILE request for MAINT at * has completed; with RC = 0.
   ```

4. Create a virtual machine with the **DIRM ADD** command and the **LIKE** parameter. In this example, the user ID is named LINUX8:

   ```
   ===> dirm add linux8 like lnxsampl pw lnx4vm
   DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
   DVHXMT1191I POKDEV62.

    DVHREQ2288I Your ADD request for LINUX76 at * has been accepted.
    ...
    DVHSHN3430I AMDISK operation for LINUX76 address 0101 has finished (WUCF
    DVHSHN3430I 07101436).
    DVHREQ2289I Your ADD request for LINUX76 at * has completed; with RC =
    DVHREQ2289I 0.
   ```

5. Allow the new user access to the virtual switches that are named VSW1 and VSW2:

   ```
   ===> rac permit system.vsw1 class(vmlan) id(linux8) access(update)
   ===> rac permit system.vsw2 class(vmlan) id(linux8) access(update)
   ```

This example shows DirMaint working with RACF when it is creating virtual machines.

10.5 Systems Management API

Note: This session is optional. Following this session only if you need Systems Management API (SMAPI).

SMAPI simplifies the task of managing many virtual images running under a single z/VM LPAR. It is a standard, platform-independent client interface that reduces the amount of required z/VM-specific programming skills.

A robust suite of Application Programming Interfaces (APIs) that perform system management functions for the z/VM Hypervisor and virtual images (guests) in a z/VM environment.

The z/VM SMAPI is the access point for any external tool to manage the z/VM running on IBMZ platform. It supports management of lifecycle and configuration of various platform resources, such as Guest, CPU, memory, virtual switches, storage, and more.

Some IBM products use the SMAPI to perform various tasks on the z/VM system. Therefore, it is necessary to make sure that SMAPI is configured and running before you configure any cloud piece that interacts with z/VM. The exact configuration steps for SMAPI might differ from the following section based on the version and release level of z/VM.

z/VM ships a set of servers that provide local system management APIs (see Figure 10-3). These servers consist of request servers that accept local connections, receive the data, and then call one of a set of worker servers to process the request. These servers are known collectively as SMAPI. The worker servers can interact with the z/VM hypervisor (CP) or with a directory manager. A directory manager is required for this environment.

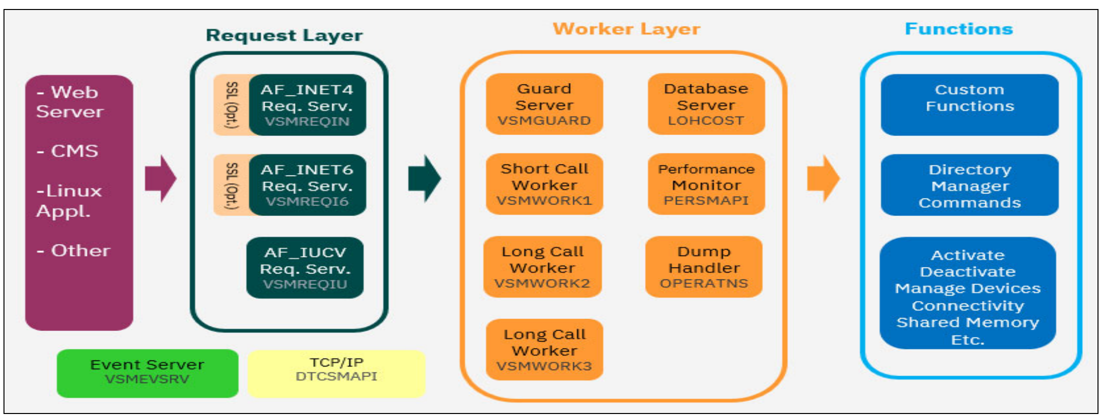


Figure 10-3 SMAPI socket-based servers work together

Note: SMAPI is a method for management software to make calls to z/VM for various operations. It is used by such products as IBM Wave and IBM Cloud Infrastructure Center, and is part of the stack of APIs that provides OpenStack operations on z/VM.

SAMPI includes a basic set of interfaces that can be used to perform the following tasks:

- Create virtual images in various operating environments:
  - IBM z/VM
  - IBM z/OS
  - IBM z/VSE

  - z/TPF
  - CMS
  - Linux on IBM Z
- Allocate, manage resources, and connectivity for virtual images
- Manage DASD volumes and groups
- Change a virtual image configuration
- Activate and deactivate individual or multiple virtual images
- Query the time when a virtual image was activated.

> **Note:** Other functions not listed here. For more information, see *z/VM: z/VM 7.2 Systems Management Application Programming Manual*.

Consider the following points when SMAPI is used:

- VSMGUARD must always be used to start SMAPI, regardless of whether the system is running in a Unified Resource Manager environment.
- A Directory Manager license is not required. If a Directory Manager is not purchased and installed, a "SMAPI USE ONLY" instance of DirMaint is installed and configured.
- A Performance Toolkit license is not required. SMAPI installs and configures a "SMAPI USE ONLY" instance of the Performance Toolkit to obtain performance data for use in provided SMAPI APIs.

10.5.1 Who needs SMAPI

SMAPI is ideal for the following user types:

- Cloud solutions, such as IBM Cloud Infrastructure Center, IBM Wave, and IBM Cloud Connector
- Any Business Partner or customer who wants to provide alternative methods or access to z/VM SMAPI functions
- Primarily for those that are running a IBM Z, but without IBM Z z/VM skills, which mostly are customers that are running Linux application on IBM Z
- Third-party solution providers

10.5.2 Configuring SMAPI to work with RACF

Complete the following steps to allow SMAPI to work with RACF. All of the following commands can be put into an EXEC, as shown in “SMAPIRAC EXEC” on page 328:

1. Access your system through a 3270 emulator.
2. Log on to `MAINT` on the first SSI member.
3. Allow `VSMWORK1` to have `CONTROL` authority of the z/VM minidisk (`VMMDISK`) that contains the `SYSTEM CONFIG` file (`PMAINT CF0`). Run the following commands:

```
===> rac permit pmaint.cf0 class(vmmdisk) acc(control) id(vsmwork1)
===> rac permit maint.cf1 class(vmmdisk) acc(control) id(vsmwork1)
```

4. Allow `VSMWORK1` to have `CONTROL` access to the generic class `VMBATCH`:

```
===> rac permit ** class(vmbatch) id(vsmwork1) access(control)
```

5. Allow SMAPI workers to read the `TCPMAINT 198` disk:

```
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmguard)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork1)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork2)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork3)
```

6. Allow `LNXADMIN` to read certain disks:

```
===> rac permit pmaint.cf0 class(vmmdisk) acc(read) id(lnxadmin)
===> rac permit autolog1.191 class(vmmdisk) acc(read) id(lnxadmin)
===> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(lnxadmin)
```

7. Change the default password expiration to your security standard. It is 90 days in this example:

```
===> rac setropts password(interval(90))
```

Enabling RACROUTE

Enable the SMAPI service machines `VSMREQI6`, `VSMREQIN`, `VSMREQIU`, `VSMEVSRV`, `DTCSMAPI`, `VSMWORK1`, `VSMWORK2`, and `VSMWORK3` to use **RACROUTE** services with the following commands:

```
===> RAC SETROPTS CLASSACT(FACILITY)
===> RAC RDEFINE FACILITY ICHCONN UACC(NONE)
ICH10006I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE ADDITION(S)
UNTIL
 A SETROPTS REFRESH IS ISSUED.
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQI6) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIN) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIU) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMEVSRV) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(DTCSMAPI) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK1) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK2) ACCESS(UPDATE)
...
===> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK3) ACCESS(UPDATE)
...
===> RAC SETROPTS RACLIST(FACILITY)
```

Exempting SMAPI from specific command checking

You must make four SMAPI service machines (`DTCSMAPI`, `VSMWORK1`, `VSMWORK2`, and `VSMWORK3`) exempt from access checking. Even if access checking is not active on your system, make the SMAPI service machines exempt from access checking for the **FOR** (privilege class C) and **LINK** commands.

Complete the following steps:

1. Make the `DTCSMAPI` virtual machine exempt by using the following commands:

```
===> RAC SETROPTS CLASSACT(VMXEVENT)
===> RAC RDEFINE VMXEVENT USERSEL.DTCSMAPI
===> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(LINK/NOCTL)
```

```
===> RAC SETEVENT REFRESH USERSEL.DTCSMAPI
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

2. Make the VSMWORK1 virtual machine exempt by using the following commands:

```
===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK1
===> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.VSMWORK1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

3. Make the VSMWORK2 virtual machine exempt by using the following commands:

```
===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK2
===> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(FOR.C/NOCTL)
===> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(LINK/NOCTL)
===> RAC SETEVENT REFRESH USERSEL.VSMWORK2
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
```

```
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

4. Make the VSMWORK3 virtual machine exempt by using the commands that are shown in Example 10-7.

Example 10-7 Commands to exempt the VSMWORK3 machine

```
   ===> RAC RDEFINE VMXEVENT USERSEL.VSMWORK3
   ===> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(FOR.C/NOCTL)
   ===> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(LINK/NOCTL)
   ===> RAC SETEVENT REFRESH USERSEL.VSMWORK3
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
   RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

SMAPIRAC EXEC

You can put all of the RACF definitions that are listed in Example 10-6 in an exec that is called SMAPIRAC (as shown in Example 10-8) to run all RACF commands at once.

Example 10-8 SMAPI RAC exec example

```
/* Edi - 2020 */
'RAC SETROPTS CLASSACT(FACILITY)'
'RAC SETROPTS RACLIST(FACILITY)'
'RAC RDEFINE FACILITY ICHCONN UACC(NONE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQI6) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIN) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIU) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMEVSRV) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(DTCSMAPI) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMGUARD) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK1) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK2) ACCESS(UPDATE)'
'RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK3) ACCESS(UPDATE)'
'RAC SETROPTS RACLIST(FACILITY) REFRESH'
```

```
'RAC RDEFINE VMXEVENT USERSEL.DTCSMAPI'
'RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(FOR.C/NOCTL)'
'RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(LINK/NOCTL)'
'RAC SETEVENT REFRESH USERSEL.DTCSMAPI'

'RAC RDEFINE VMXEVENT USERSEL.MAINT'
'RAC RALTER VMXEVENT USERSEL.MAINT ADDMEM(FOR.C/NOCTL)'
'RAC RALTER VMXEVENT USERSEL.MAINT ADDMEM(LINK/NOCTL)'
'RAC SETEVENT REFRESH USERSEL.MAINT'

'RAC RDEFINE VMCMD DIAG088 UACC(NONE)'
'RAC SETROPTS CLASSACT(VMCMD)'

'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQIN) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQI6) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQIU) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMEVSRV) ACCESS(READ)'

'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMGUARD) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK1) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK2) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK3) ACCESS(READ)'

'RAC PERMIT DIAG088 CLASS(VMCMD) ID(LOHCOST) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(DTCSMAPI) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(PERSMAPI) ACCESS(READ)'
'RAC PERMIT DIAG088 CLASS(VMCMD) ID(OPERATNS) ACCESS(READ)'

'RAC PERMIT MAINT720.5E5 CLASS(VMMDISK) ID(VSMWORK1) ACCESS(READ)'
'RAC PERMIT MAINT720.51D CLASS(VMMDISK) ID(VSMWORK1) ACCESS(READ)'
'RAC PERMIT PMAINT.551 CLASS(VMMDISK) ID(VSMGUARD) ACCESS(READ)'

'RAC PERMIT PMAINT.CF0 CLASS(VMMDISK) ACC(CONTROL) ID(VSMWORK1)'
'RAC PERMIT MAINT.CF1 CLASS(VMMDISK) ACC(CONTROL) ID(VSMWORK1)'

'RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMGUARD)'
'RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK1)'
'RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK2)'
'RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK3)'

'RAC PERMIT MAINT CLASS(VMRDR) ID(DTCSMAPI) ACCESS(UPDATE)'
'RAC PERMIT TCPMAINT CLASS(VMRDR) ID(DTCSMAPI) ACCESS(UPDATE)'

'RAC PERMIT DIRMAINT CLASS(VMRDR) ID(VSMWORK2) ACCESS(UPDATE)'
'RAC PERMIT DIRMAINT CLASS(VMRDR) ID(VSMWORK3) ACCESS(UPDATE)'

'RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK1) ACCESS(CONTROL)'
'RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK2) ACCESS(CONTROL)'
'RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK3) ACCESS(CONTROL)'
'RAC PERMIT ** CLASS(VMBATCH) ID(DTCSMAPI) ACCESS(CONTROL)'

'RAC PERMIT CLASS(VMBATCH) ID(VSMWORK1) ACCESS(CONTROL)'
'RAC PERMIT CLASS(VMBATCH) ID(VSMWORK2) ACCESS(CONTROL)'
'RAC PERMIT CLASS(VMBATCH) ID(VSMWORK3) ACCESS(CONTROL)'
```

```
'RAC PERMIT CLASS(VMBATCH) ID(DTCSMAPI) ACCESS(CONTROL)'
```

RACF can now allow SMAPI to do its job. It is recommended that you follow the instructions in “Test SMAPI from Linux by using smaclient.” on page 333, and 10.5.10, “Testing SMAPI from Linux by using smaclient” on page 339.

10.5.3 Shared File System that is used by SMAPI

The SMAPI request servers and worker servers use Shared File System (SFS) directories to access configuration files and other data. SMAPI uses the standard file pool VMSYS and VMPSFS to keep their files. The VSMWORK1 user ID is the owner of some of the SFS directories that have control files, logs, and so on.

The SFS directories are defined on SFS file pools. The authorization and ownership for the SFS directories are done by using enroll SFS commands. You can set AUDIT parameter on DSMPARMS file for auditing purpose.

> **Note:** For more information about managing and auditing the VMSYS or VMPSFS file pools, see *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261.

All commands that are shown in this chapter regarding SFS ENROLL and GRANT are performed automatically during z/VM installation. They are shown here for verification and testing purposes.

Also, if you are adding a worker or request server, you can use the appropriate commands from these lists as a guide for enrolling your new server in the correct file pool and then grant SFS authorizations.

> **Note:** You can run the steps that are shown in Example 10-9 or create an EXEC, as described in **“SMAPISFS EXEC” on page 331**.

Example 10-9 SFS ENROLL command to SMAPI userids

```
ENROLL USER VSMWORK1 VMSYS: (BLOCKS 6000 STORGROUP 2
ENROLL USER VSMWORK2 VMSYS:
ENROLL USER VSMWORK3 VMSYS:
ENROLL USER VSMREQIN VMSYS:
ENROLL USER VSMREQIU VMSYS:
ENROLL USER VSMGUARD VMPSFS: (BLOCKS 1000 STORGROUP 2
ENROLL USER VSMGUARD VMSYS:
ENROLL USER VSMREQI6 VMSYS:
ENROLL USER VSMEVSRV VMSYS:
ENROLL USER DTCSMAPI VMSYS:
ENROLL USER OPERATNS VMSYS:
ENROLL USER PERSMAPI VMSYS: (BLOCKS 24000 STORGROUP 2
ENROLL USER OPNCLOUD VMSYS:
```

If you do not grant access to the specific directory, you cannot access it. Example 10-10 shows SFS GRANT commands that are automatically performed during z/VM installation.

Example 10-10 SFS GRANT command to SMAPI userids

```
GRANT AUTHORITY VMSYS:VSMWORK1. TO OPNCLOUD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO OPNCLOUD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1. TO MAINT (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO MAINT (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1. TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK2 (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK3 (WRITE NEWWRITE
GRANT AUTHORITY * * VMSYS:VSMWORK1. TO VSMGUARD (READ
GRANT AUTHORITY VMSYS:VSMWORK1. TO PERSMAPI (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMAINT (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT2 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT3 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT4 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOVE (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV2 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV3 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV4 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG1 (WRITE NEWWRITE
GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG2 (WRITE NEWWRITE
```

SMAPISFS EXEC

You can create an EXEC similar to the example that is shown in Example 10-11 to run multiple commands to enroll multiple users or servers in a file pool and create an SFS type file space.

Example 10-11 SFS commands to define SMAPI user IDs

```
/* Edi 2020 */
'ENROLL USER VSMWORK1 VMSYS: (BLOCKS 6000 STORGROUP 2'
'ENROLL USER VSMWORK2 VMSYS:'
'ENROLL USER VSMWORK3 VMSYS:'
'ENROLL USER VSMREQIN VMSYS:'
'ENROLL USER VSMREQIU VMSYS:'
'ENROLL USER VSMGUARD VMPSFS: (BLOCKS 1000 STORGROUP 2'
'ENROLL USER VSMGUARD VMSYS:'
'ENROLL USER VSMREQI6 VMSYS:'
'ENROLL USER VSMEVSRV VMSYS:'
'ENROLL USER DTCSMAPI VMSYS:'
'ENROLL USER OPERATNS VMSYS:'
'ENROLL USER PERSMAPI VMSYS: (BLOCKS 24000 STORGROUP 2'
'GRANT AUTHORITY VMSYS:VSMWORK1. TO MAINT (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO MAINT (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1. TO VSMGUARD (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO VSMGUARD (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMGUARD (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK2 (WRITE NEWWRITE'
'GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK3 (WRITE NEWWRITE'
```

```
'GRANT AUTHORITY * * VMSYS:VSMWORK1. TO VSMGUARD (READ'
'GRANT AUTHORITY VMSYS:VSMWORK1. TO PERSMAPI (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMAINT (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT2 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT3 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT4 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOVE (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV2 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV3 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV4 (READ NEWREAD'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG1 (WRITE NEWWRITE'
'GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG2 (WRITE NEWWRITE'
```

10.5.4 SMAPI requirements

Tip: A directory manager is required that can be IBM z/VM DirMaint or another vendor product.

z/VM DirMaint 7.2 or later is required to support the new socket-based environment that consists of one or more request servers and two or more worker servers. If using different directory manager, some exit replacement is needed.

Refer to chapter 3 of z/VM: z/VM 7.2 Systems Management Application Programming (SMAPI) manual.

10.5.5 Configuring DirMaint to support SMAPI

Complete the following steps:

1. Log on MAINT720 user ID.
2. Create the DirMaint configuration file, `CONFIGSM DATADVH L`. The `L` disk is on `7VMDIR20 41F`, which is the preproduction disk. Add the following lines:

```
===> vmlink 7VMDIR20 41f <41f L> (w
===> x configsm datadvh a
===> a 10
RUNMODE= OPERATIONAL
ONLINE=  IMMED
ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
ALLOW_ASUSER_NOPASS_FROM= PERSMAPI *
ALLOW_ASUSER_NOPASS_FROM= LOHCOST *
ALLOW_ASUSER_NOPASS_FROM= DTCSMAPI *
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP=  DVHXNE   EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP=  DVHXNE   EXEC
```

Notes: Consider the following points:

- The `ALLOW_ASUSER_NOPASS_FROM` lines allow SMAPI users to issue commands to the Directory Manager by using the `ASUSER` modifier and the password of that user.
- The `ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT` lines activate an exit that notifies SMAPI of changes that are made to the user directory.

3. Add the SMAPI user IDs in the `AUTHFOR CONTROL` file. As DirMaint is up and running after all steps are completed, you must request a copy of this file from DirMaing by running the following command:

   ```
   ===> dirm send AUTHFOR CONTROL
   ```

   This file is sent by DirMaint to your reader; then, run the following command to receive this file to your 191 MDisk accessed as A:

   ```
   ===> Readerlist or RL
   ```

4. Position the cursor over the line of AUTHFOR CONTROL is and issue PF9 to receive it. Then, issue filelist to xedit it and include the SMAPI user IDs that have authorization to issue DirMaint commands: `VSMGUARD`, `VSMWORK1`, `VSMWORK2`, and `VSMWORK3`:

   ```
   ===> x authfor control a
   ===> a 10
   ALL VSMGUARD * 140A ADGHOPS
   ALL VSMGUARD * 150A ADGHOPS
   ALL VSMWORK1 * 140A ADGHOPS
   ALL VSMWORK1 * 150A ADGHOPS
   ALL VSMWORK2 * 140A ADGHOPS
   ALL VSMWORK2 * 150A ADGHOPS
   ALL VSMWORK3 * 140A ADGHOPS
   ALL VSMWORK3 * 150A ADGHOPS
   ===> file
   ```

5. Send the files you create to DirMaint and reload the file:

   ```
   ===> dirm file CONFIGSM DATADVH
   ===> dirm file AUTHFOR CONTROL
   ===> dirm rldd
   ```

DirMaint configuration files for SMAPI were created and activated. After DirMaint (or another directory maintenance product) is configured, SMAPI can be enabled and configured. To set up SMAPI, perform the following tasks:

1. Define SMAPI on RACF.
2. Test SMAPI from the Conversational Monitor System (CMS).
3. Test SMAPI from Linux by using smaclient.

10.5.6 Setting up basic SMAPI configuration

Complete the following steps on only one SSI member:

1. Log on to `MAINT` on SSI member 1.
2. Grant authority to the `VSMGUARD` virtual machine to use certain SFS directories with the following three **`GRANT`** commands:

```
===> grant authority vmsys:vsmwork1. to vsmguard (write newwrite
===> grant authority vmsys:vsmwork1.data to vsmguard (write newwrite
===> grant authority * * vmsys:vsmwork1. to vsmguard (read
```

3. Access the SFS `VMSYS:VSMWORK1` as your `F` disk in read/write mode:

```
===> access vmsys:vsmwork1. f (forcerw
```

4. Edit the file `VSMWORK1 AUTHLIST` on that disk:

```
===> x vsmwork1 authlist f
```

5. Duplicate the last line by putting a double quotation mark (“) in the prefix area:

> **Note**: It is important to duplicate the line because lines must be 195 characters wide.

```
00001 DO.NOT.REMOVE
DO.NOT.REMOVE
00002 MAINT                                                        ALL
00003 VSMPROXY                                                     ALL
" 004 ZVMLXAPP                                                      ALL
```

6. Press **Enter** and the line is duplicated. Replace the user ID with `LNXADMIN` and save the file:

```
00001 DO.NOT.REMOVE
DO.NOT.RE
MOVE
00002 MAINT                                                        ALL
00003 VSMPROXY                                                     ALL
00004 ZVMLXAPP                                                     ALL
00005 LNXADMIN                                                      ALL
```

This change allows the `LNXADMIN` virtual machine to invoke SMAPI calls.

10.5.7 Defining SMAPI on RACF

For more information, see 6.7.7, “Configuring SMAPI to work with RACF” on page 173.

10.5.8 Start SMAPI at IPL time

Complete the following steps to start SMAPI at IPL time by adding one line to the **PROFILE EXEC** on the AUTOLOG1 191 disk:

1. Link the AUTOLOG1 191 disk read/write and access it as file mode I:

```
===> link autolog2 191 1191 mr
DASD 1192 LINKED R/W;
===> acc 1191 t
```

2. Edit the PROFILE EXEC and add one line to start SMAPI:

```
===> x profile exec t
...
/*******************************************************************/
/* Customer processing can be added here                           */
/*******************************************************************/
"CP XAUTOLOG TCPIP"             /* Start TCPIP                   */
"PIPE CP SLEEP 30 SEC"
"CP XAUTOLOG LNXADMIN"          /* Start the Linux admin machine    */
"CP XAUTOLOG VSMGUARD"          /* Start SMAPI                     */
...
```

3. Repeat this process for all other members in the SSI cluster.

Verifying that SMAPI comes up at IPL time

Complete the following steps to verify that SMAPI comes up after an IPL:

> **Note:** If you downloaded the execs from the FTP serve, you can use SSICMD as shown here or use CP commands.

1. Query the virtual machines that are running by using the **SSICMD EXEC** and the **QUERY NAMES** command to query all active virtual machines on all members or use the CP command **Q N AT ALL**:

```
===> ssicmd q n
RDBKZVMH:
DIRMSAT2 - SSI
FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMAINT - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC
VMSERVU  - DSC , VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP     - DSC , OPERATOR - DSC , MAINT    -L0004
VSM     - TCPIP

RDBKZVMG:
VMSERVP  - SSI , DIRMAINT - SSI
FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMSAT2 - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC
VSM     - TCPIP

===> q n at all
RDBKZVMH : TCPIP    - DSC , DTCVSW4  - DSC , DTCVSW3  - DSC , DTCVSW2  - DSC
RDBKZVMH : DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC
RDBKZVMH : OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
```

```
RDBKZVMH : VSM      - TCPIP
RDBKZVMG : MAINT720 -L0004, DTCVSW4  - DSC , DTCVSW3  - DSC , DTCVSW2  - DSC
RDBKZVMG : DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
RDBKZVMG : VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
RDBKZVMG : TCPIP    - DSC , MAINT    -L0003
RDBKZVMG : VSM      - TCPIP
```

2. If you are sure that you are in a position to shut down, shut down and re-IPL the SSI cluster:

```
===> ssicmd shutdown reipl
or
===> cp shutdown reipl
SYSTEM SHUTDOWN STARTED
HCPSHU960I System shutdown may be delayed for up to 630 seconds
VMSERVP : DMS5BC3108I Shutdown Signal received. STOP processing started
VMSERVU : DMS5BC3108I Shutdown Signal received. STOP processing started
...
```

3. When the SSI cluster comes back up, log on as `MAINT` to the first SSI member.
4. Query the virtual machines by running with the **SSICMD EXEC** as a reference (the SMAPI virtual machines are shown in bold):

```
===> ssicmd q n
or
===> cp q n at all
RDBKZVMH:
DIRMSAT2 - SSI
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
LNXADMIN - DSC , TCPIP    - DSC , DIRMAINT - DSC , DTCVSW2  - DSC
DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC , LOHCOST  - DSC , VSMEVSRV - DSC , VSMPROXY - DSC
VSMREQIU - DSC , VSMREQI6 - DSC , VSMREQIN - DSC , DTCSMAPI - DSC
PERSMAPI - DSC , VSMWORK3 - DSC , MAINT    -L0004
VSM     - TCPIP

RDBKZVMG:
DIRMAINT - SSI , VMSERVP  - SSI
LOHCOST  - DSC , VSMEVSRV - DSC , VSMPROXY - DSC , VSMREQIU - DSC
VSMREQI6 - DSC , VSMREQIN - DSC , DTCSMAPI - DSC , PERSMAPI - DSC
VSMWORK3 - DSC , VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC
VSMGUARD - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMSAT2 - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC
VSM     - TCPIP
```

SMAPI is now running and configured.

10.5.9 Testing SMAPI from the Conversational Monitor System

The following methods can be used to test SMAPI when logged on MAINT user ID:

- SMTATUS exec
- CALLSM1 exec

These methods are described next

Using SMSTATUS exec

SMSTATUS is a special stand-alone EXEC that captures data regarding the status of the various SMAPI servers and system settings that are useful for investigating suspected problems involving SMAPI. Use it to perform the same function as SMAPI_Status_Capture when that API cannot be executed because SMAPI is not responsive.

To use this EXEC, complete the following steps:

1. The SMSTATUS EXEC is designed to be run by MAINT. Complete the following steps to run the exec:
   a. Log on as MAINT.
   b. b. Access the `vmsys:vsmwork1.data` directory:
      ```
      set filepool vmsys
      acc vmsys:vsmwork1.data f
      ```
   c. Access MAINT's 193 disk as G. It must be accessed in your search order after the `vmsys:vsmwork1.data` directory.
   d. Enter `SMSTATUS`.
2. Running SMSTATUS might prompt you for a password to test that the directory manager is configured correctly. You are prompted to check whether you are in a VMREAD state. If you are, enter your log on password to continue.
3. When the SMSTATUS EXEC completes, an output file is created in the `VMSYS:VSMWORK1.STATUS` directory, as specified by the **`Server_STATUS = attribute`** in the `DMSSICNF COPY` file. At the end of the execution, the EXEC indicates the name and location of this file:

```
q accessed
Mode  Stat     Files  Vdev   Label/Directory
A      R/W        26  191    MNT191
B      R/O       139  5E5    MNT5E5
C      R/W         4  2CC    MNT2CC
D      R/O       210  51D    MNT51D
E      R/O        10  551    PMT551
F      R/W        19  DIR    VMSYS:VSMWORK1.DATA
G      R/O      1144  193    MNT193
H      R/O        56  100    DRM11F
S      R/O       693  190    MNT190
V      R/W         2  DIR    VMSYS:VSMWORK1.STATUS
X      R/O         2  120    VW1191
Y/S    R/O      1122  19E    MNT19E
Z      R/O       169  400    MNT400
```

Usage notes

Consider the following points:

- SMSTATUS does not clear or rotate logs after it collects them.
- If you are running an External Security Manager (ESM), SMSTATUS can fail to collect console logs, even if you configured SMAPI as described in Appendix F, “Using SMAPI with an External Security Manager,” or *z/VM 7.2 Systems Management Application Programming*.

  This failure is noted in the SMSTATUS output.
- The SMSTATUS output for some ESM-related problems can be incomplete. To diagnose a problem that is related to an ESM, you might need to provide all relevant profiles, all group membership for groups that are authorized by those profiles, and the list of all groups of which any user that is listed in those profiles (directly or by way of another group) is a member.

  One way to provide this information is by using a data base dump.

Using CALLSM1 exec

To test SMAPI, a REXX EXEC that is named **CALLSM1** is included with the files that are associated with this book in Appendix C, “Additional material” on page 489. You copied the REXX EXEC **CALLSM1** to the MAINT 191 (A) disk as described in 6.11.2, “Copying the utilities to Shared File System file pools” on page 186.

If the REXX EXEC **CALLSM1** was not copied, you need to copy it to complete this section.

To test SMAPI, complete the following steps:

1. Log on to MAINT on member 1.
2. Verify that the **CALLSM1 EXEC** was copied to the MAINT 191 disk:

   ```
   ===> listfile callsm1 *
   CALLSM1  EXEC     A1
   ```
3. Link to the TCPMAINT 592 disk:

   ```
   ===> vmlink tcpmaint 592
   DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
   ```
4. Run the **CALLSM1 EXEC** with the following command. The output is shown in Example 10-12:

   ```
   ===> callsm1
   ```

Example 10-12 Output of CALLSM1 EXEC

```
buffLen = 57
0000 00000035 00000019 496D6167 655F4465 *    5    Image_De *
0016 66696E69 74696F6E 5F517565 72795F44 * finition_Query_D *
0032 4D000000 00000000 00000000 054D4149 * M            MAI *
0048 4E540000 00032A20 00                * NT    *          *

calling send()
receiving requestId,  buffLen = 4
returned from recv() rc,retvalue =0,4
Request id:= 3756453462

receiving length, buffLen = 4
returned from recv() rc,retvalue =0,4
receiving data, buffLen = 2808
```

```
returned from recv() rc,retvalue =0,2808

Request id: 3756453462 Return code:0 Reason code:0 possible outdata len:2792

<COMMAND_DEFINE_CPU=>
<COMMAND_SET_CPUAFFINITY=>
<COMMAND_SET_SHARE=>
<COMMAND_SET_VCONFIG=>
<CONSOLE=VDEV=0009 DEVTYPE=3215 CLASS=T>
...
<VMRELOCATE=>
```

The output shows that SMAPI is working from CMS.

10.5.10 Testing SMAPI from Linux by using smaclient

The script **smaclient** is a powerful, open source bash wrapper around SMAPI. It is available at this web site.

To test SMAPI by using **smaclient**, complete the following steps:

1. Start a root SSH session on the Linux system that is running on one LNXADMIN.
2. If your Linux system can access the internet, you can get the script directly by using the **wget** command:

```
# cd /usr/local/sbin
# wget http://download.sinenomine.net/smaclient/smaclient-1.1
--2013-06-13 09:55:22--  http://download.sinenomine.net/smaclient/smaclient-1.1
...
2013-06-13 09:55:22 (3.20 MB/s) - `smaclient-1.1' saved [332722/332722]
# mv smaclient-1.1 smaclient
```

3. If your Linux system cannot access the internet, complete the following steps:
   a. Download the script from the previous URL to a workstation.
   b. Upload the script from the workstation to one of the LNXADMIN systems to the file directory /usr/local/sbin/smaclient.
4. Make the script executable with the **chmod +x** command and verify that it is in the root's path by using the **which** command:

```
# chmod +x smaclient
# which smaclient
/usr/local/sbin/smaclient
```

5. Create the file /etc/smaclient.conf so that inter-user communication vehicle (IUCV) is used to communicate to SMAPI:

```
# cd /etc
# vi smaclient.conf
smhost="IUCV"
```

6. Build the **smiucv** binary with the following command. To build it, ensure that the GNU collection of compilers (gcc) is installed:

```
# smaclient smiucv
smiucv built as /usr/local/sbin/smiucv
```

Ensure that `/usr/local/sbin` is included in PATH.

If gcc is not installed, you might first need to run the command **`yum install gcc`** on RHEL, or **`zypper install gcc`** on SLES.

7. Test a SMAPI call by using **`smaclient`**. The argument **`Image_Query_DM`** in the following command calls the SMAPI that queries a user directory entry, in this example, `LNXADMIN`:

```
# smaclient Image_Query_DM -T lnxadmin
IDENTITY LNXADMIN LNX4VM 512M 4G BDEG
06130733
   INCLUDE LNXDFLT
06130733
   BUILD ON ITSOZVM1 USING SUBCONFIG LNXADM-1
06130733
   BUILD ON ITSOZVM2 USING SUBCONFIG LNXADM-2
06130733
   IUCV ANY
06130733
   OPTION MAXCONN 128 LNKNOPAS
06130733
...
```

This output shows that SMAPI is running, `LNXADMIN` is correctly authorized to call SMAPI, and the Linux interface **`smaclient`** is working.

10.6 Adding a z/VM user ID

To add a user to z/VM, you must create a directory entry for each new virtual machine. The default method is through a manual process where you update a file that is called USER DIRECT, which is the z/VM system directory. After installing DirMaint, it will be doing the user ID and resource management on z/VM directory.

> **Tip:** The USER DIRECT file is a CMS file that contains the configuration values and the definition settings for each virtual machine (known as z/VM user ID). It acts as a catalog of user IDs that can log on to z/VM, including Linux servers.

A virtual machine definition is a grouping of directory statements that begin with the term USER or IDENTITY as shown in Example 10-13. For more information, a description of virtual machine types, and the USER and IDENTITY statements, see Chapter 1 of *z/VM Getting Started with Linux on System z*, SC24-6194.

Example 10-13 z/VM user ID example

```
USER DIRMAINT AUTOONLY  128M  256M BDG
 IPL CMS PARM AUTOCR
 MACHINE ESA
 ACCOUNT SYSTEM SYSPROG
 D8ONECMD FAIL LOCK
 OPTION CONCEAL DIAG88 D84NOPAS IGNMAXU
 IUCV ALLOW PRIORITY MSGLIMIT 100
 CONSOLE 009 3215 T
 SPOOL 00C 2540 READER A
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
```

```
LINK 7VMDIR20 491 191 MR
LINK 7VMDIR20 492 192 MR
LINK 7VMDIR20 11F 11F MR
LINK 7VMDIR20 41F 21F MR
LINK MAINT 190 190 RR * CMS system disk
LINK MAINT 19D 19D RR * help disk
LINK MAINT 19E 19E RR * Product code disk
LINK MAINT 123 123 MW
LINK PMAINT 551 551 RR
LINK TCPMAINT 592 592 RR
MDISK 1AA 3390 06160 020 VMBCM1 MR
MDISK 1FA 3390 06180 012 VMBCM1 MR
MDISK 1DE 3390 06192 020 VMBCM1 MR
MDISK 2AA 3390 06212 020 VMBCM1 MR
MDISK 155 3390 06232 012 VMBCM1 MR
MDISK 1DF 3390 06244 012 VMBCM1 MR
MDISK 1DB 3390 06256 012 VMBCM1 MR
MDISK 2DF 3390 06268 012 VMBCM1 MR
MDISK 2DB 3390 06280 012 VMBCM1 MR
MDISK 15D 3390 06292 009 VMBCM1 MR
```

10.6.1 DirMaint commands

To add a USER or IDENTITY, you must enter the DirMaint commands to include this definition in the z/VM directory.

A HELP menu is available in z/VM to clarify any commands that you want to issue. This function shows different levels of information that is useful and can help the z/VM administrator to better understand the commands.

Tip: The following options are available on the HELP MENU:

- DirMaint Commands
- DirMaint Topics

To see the DirMaint commands, position the cursor over an option and press **Enter**.

DirMaint commands overview

The DirMaint command must be preceded by the abbreviation DIRM. This abbreviation routes the command to the DIRMAINT service machine, where the service machine performs validation checking and processes the request or rejects it with a suitable message.

Quick reference for the most useful commands

In this section, we present a few of the most useful DirMaint commands.

General management and administration

`dirm direct`	Places the current directory online
`dirm enable`	Places the DirMaint in an enabled state
`dirm backup`	Creates a USER BACKUP file on DIRMAINT 1DB MDisk
`dirm user withpass`	Get a copy of USER DIRECT with passwords
`dirm user nopass`	Get a copy of USER DIRECT without passwords

dirm query dvhlevel	Get the DirMaint release and maintenance applied
dirm status workunit all	Get the list of pending work units
dirm status datamove all	Get the list of pending work units
dirm cms q disk	Get the info of which MDisks are accessed by DIRMAINT
dirm status locked	Get a list of userids locked

Note: These shutdown commands are disruptive.

dirm dat shutdown	Shutdown the datamove user ID
dirm sat shutdown	Shutdown the dirmsat user ID
dirm shutdown	Shutdown the dirmaint user ID

User ID and disk management

dirm add <userid>	Add a new user ID direct
dirm for <userid> dir for <userid> setpw lbyonly	Modify the user ID's password to restricted access
dirm for <userid> get	Receive a copy of user ID direct on your reader in order you can modify any user ID definitions
dirm for <userid> replace	Replace the modified copy of user ID direct
dirm for <userid> purge	Purge a user ID and all its resources defined on directory
dirm for <userid> amd	Add an MDisk on the user ID direct
dirm for <userid> cmd	Change the MDisk size or location
dirm for <userid> dmd	Delete MDisk, this MDisk is deleted immediately
dirm status workunit	Status of a workunit (request)
dirm workunit <number> retry	Retry a workunit (request)
dirm workunit <number> wakeup	Wakeup a request (workunit)
dirm workunit <number> cancel	Cancel a request (workunit)

DASD management

`dirm dasd query group *` Get the DASD groups defined on EXTENT CONTROL

`dirm freext` Get all DASD free space list

`dirm usedext` Get all DASD used space list

For more information, see 11.4, “Common DirMaint tasks” on page 362.

11

Deploying and maintaining Linux workloads

This chapter describes how to begin your first Linux installation. It also describes common tasks that are run during administration, maintenance, and expansion tasks to accommodate more workloads.

This chapter includes the following topics:

11.1 Planning a Linux virtual machine

Every server needs the following minimal set of items to operate:

- CPU
- Memory (the mainframe term is *storage*; the Linux term is typically *RAM*)
- Disk
- Network

z/VM has several different options to provide this set of items, which is typically a mix of physical and virtual hardware. Your selections influence the cost and behavior of the server that is generated. The following sections show the differences among the available hardware types and cover procedures to implement or configure each type. The Linux virtual machine definitions are out for scope of this chapter.

11.2 Considerations for disk storage types

The type of disk to use often depends on the actual configuration of the mainframe. The available hardware types are count key data (CKD) disks with a Fibre Channel connection (FICON), and Fibre Channel (FC) disks. Several methodologies are possible to configure both types.

11.2.1 Direct-attached storage devices (DASD)

Extended count key data (ECKD) DASD is the traditional disk storage hardware for use on the mainframe. This type of disk is available on all mainframe systems that run z/OS, IBM z/VSE, z/TPF, or z/VM with the IBM z/VM Single System Image (VMSSI) feature enabled. This type of disk is advantageous when you use large amounts of disk, and it scales well with the number of disks. Therefore, by giving a small amount of DASD to each of the virtual machines, each virtual machine can performance well.

DASD offers these advantages:

- Simplistic allocation with a directory maintenance product.
- Easy dedication of disks by a 4-digit hexadecimal number.
- Good scalability over thousands of disks.
- Simplified setups for disaster recovery.
- Disk sizes are multiples of a Model 1.
- High-performance storage systems are available for this type of disks.
- Capability to use ultra-high-speed FlashCopy feature if it is present in the storage subsystem.

DASD has limitations:

- Only one I/O operation occurs at a time on a single DASD device, which can be improved with parallel access volume (PAV) or HyperPAV.
- Limited maximum size of disks.
- Disks cannot be resized easily.

All CKD disks are identified by a 4-digit hexadecimal number within a logical partition (LPAR).

Table 11-1 on page 347 shows the standard 3390 DASD models.

Table 11-1 Standard 3390 DASD models

Model	Cylinders	Storage capacity
Model-3	3,339	2.83 GB
Model-9	10,017	8.51 GB
Model-27	32,760	27.84 GB
Model-54	65,520	55.68 GB

Note: The Model-27 DASD and the Model-54 DASD are not multiples of Model-9. The actual Model-54 size is debated, but it helps to know that Model-54 came with DS8000. Model-54 is configured with DS8000 as 3390-A. Model-27 is considered half of the size of a Model-54.

Using dedicated DASD for Linux virtual machines

When you use DASD, it is helpful to track all of the different DASD numbers and assignments by using the planning worksheet in Appendix B, "Reference, cheat sheets, blank worksheets, and education" on page 475.

Although use cases exist for directly dedicating DASD to an individual virtual machine, the preferred method is to add a full-pack minidisk to the virtual machine instead. To attach a DASD to a z/VM virtual machine directly, use the **DIRMAINT** command, as shown in Example 11-1. DirMaint automatically places the entry at the correct location in the directory entry for you.

Example 11-1 Dedicate DASD number 1570 with virtual address 0705

```
===> dirmaint foruser linux2 dedicate 0705 1570
DVHXMT1191I Your DEDICATE request has been sent for processing to
DVHXMT1191I DIRMAINT at ITSOZVM1.
Ready; T=0.01/0.01 13:03:47
 DVHREQ2288I Your DEDICATE request for LINUX2 at * has been accepted.
 DVHBIU3450I The source for directory entry LINUX2 has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHDRC3451I The next ONLINE will take place via delta object directory.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry LINUX2 have been placed online.
 DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
```

To manually dedicate the same DASD without requiring the virtual machine to log on again, use the following command:

```
===> ATTACH 1570 TO LINUX2 AS 0705
```

The ATTACH command is only temporary. The directory change by using DirMaint is still necessary to attach the dedicated DASD to the virtual machine persistently.

If the DASD subsystem supports the creation of multiple aliases to the same device (HyperPAV), it is possible to use a real volume's alias devices to run more than one virtual machine I/O at a time on the volume. This capability helps to prevent I/O operations from blocking one another, which can result in virtual machine I/O operations with reduced response time, which in turn allows the workload on virtual machines to run more quickly. For more information about HyperPAV, see 11.2.5, "HyperPAV" on page 358.

11.2.2 Direct-attached Fibre Channel

Storage area networks (SANs) are specialized networks that are dedicated to the transport of mass storage data. Today, the most common SAN technology is Fibre Channel. FC is established in nearly all large enterprise-class data centers globally. On the mainframe, the protocol is called *Fibre Channel Protocol* (FCP). It is important to distinguish FCP from mainframe architecture, where the abbreviation FC stands for FICON. Although FCP and FC are similar, they are not the same.

Direct-attached Fibre Channel characteristics

FCP features the following specific properties:

- No disks are dedicated, only adapters. The Linux operating system operates the Fibre Channel adapters, as shown in Figure 11-1.

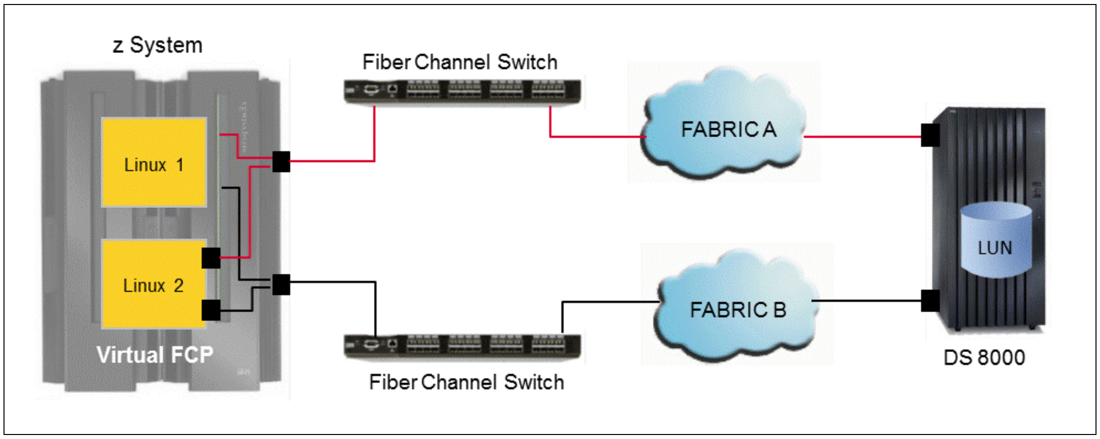


Figure 11-1 Connection overview between disk storage subsystem and virtual machines

- Each adapter can provide many disks, which are also called *logical unit numbers* (LUNs). The exact number depends on the storage system that is used.
- When you use N_Port ID Virtualization (NPIV) as recommended in this book, the Fibre Channel switches must support the NPIV protocol.
- Fibre Channel disks can be created with enormous volume sizes, which are exponentially larger than any single disk that is available today.
- Because Fibre Channel uses Small Computer System Interface (SCSI) commands over fiber-optic cable, it also uses SCSI command queuing, which results in multiple commands that are processed at the same time.
- Fibre Channel storage systems are available from many vendors in many price ranges.
- Because the Linux OS that is running inside of the virtual machine must manage the adapters, it also must manage availability. Configuring dual adapters with multipathing by using dual fabrics is considered mandatory by the authors of this book.
- The setup of the disks in the disk storage subsystem and the zoning of the adapters involve many worldwide port numbers (WWPNs). Meticulous attention to detail is critical to a successful outcome and special care must be taken.
- For IBM Z, any supported FCP adapter, such as FICON Express or FICON Express2, can be used for this purpose.

You must track the FCP adapters like the DASD that is available to the system. Although it is not possible to attach the same FCP adapter to two different z/VM virtual machine systems, it is important to maintain accurate records that show the adapter that is dedicated to each virtual machine.

Configuring z/VM to use direct-attached Fibre Channel

To be able to relocate a running Linux virtual machine with FCP devices from one LPAR to another, FCP device numbers on each LPAR must be the same and the equivalency identifier (EQID) must be set up.

To create an EQID dynamically, run the **SET RDEVICE** command on each single system image (SSI) LPAR where Linux needs to relocate. Complete the following steps:

1. Log on as MAINT.
2. Vary devices offline:

```
===> vary off b801-b802 ba01-ba02
BA01 varied offline
BA02 varied offline
BB01 varied offline
BB02 varied offline
4 device(s) specified; 4 device(s) successfully varied offline
```

3. Use **SET RDEVICE** to create EQIDs dynamically:

```
===> set rdev b801 eqid fcpid01 type fcp
HCPZRP6722I Characteristics of device B801 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev b901 eqid fcpid01 type fcp
HCPZRP6722I Characteristics of device B901 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev b802 eqid fcpid02 type fcp
HCPZRP6722I Characteristics of device B802 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev b902 eqid fcpid02 type fcp
HCPZRP6722I Characteristics of device B902 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
```

4. Check the result with the **QUERY EQID** command:

```
===> query eqid fcpid01
Devices for FCPID01:
B801 B901
===> query eqid fcpid02
Devices for FCPID02:
B802 B902
```

5. Vary the devices online with the **VARY ON** command:

```
===> vary on b801-b802 b901-b902
B801 varied online
B802 varied online
B901 varied online
B902 varied online
4 device(s) specified; 4 device(s) successfully varied online
```

6. Repeat these steps on the other nodes of the SSI.

To make the EQIDs permanent, complete the following steps:

1. Edit the `SYSTEM CONFIG` file and add **RDEV** statements:

```
===> vmlink pmaint cf0 < cf0 f mr >
===> xedit system config f
/* Add EQID statements for OSA addresses, unique MAC IDs and FCP*/
ZVM63A: BEGIN
  RDEV 2100-210F EQID OSA1SET1 TYPE OSA
  RDEV 2120-212F EQID OSA1SET1 TYPE OSA
  VMLAN MACPREFIX 02000B
  VMLAN LIMIT TRANSIENT 0
  DEFINE VSWITCH VSW1 RDEV 2103 2123 ETHERNET
  DEFINE VSWITCH VSW2 ETHERNET
  RDEV B801 EQID FCPID01 TYPE FCP
  RDEV B802 EQID FCPID02 TYPE FCP
  RDEV B901 EQID FCPID01 TYPE FCP
  RDEV B902 EQID FCPID02 TYPE FCP
ZVM63A: END
ZVM63B: BEGIN
  RDEV 2040-204F EQID OSA1SET1 TYPE OSA
  RDEV 2060-206F EQID OSA1SET1 TYPE OSA
  VMLAN MACPREFIX 02000C
  VMLAN LIMIT TRANSIENT 0
  DEFINE VSWITCH VSW1 RDEV 2043 2063 ETHERNET
  DEFINE VSWITCH VSW2 ETHERNET
  RDEV B801 EQID FCPID01 TYPE FCP
  RDEV B802 EQID FCPID02 TYPE FCP
  RDEV B901 EQID FCPID01 TYPE FCP
  RDEV B902 EQID FCPID02 TYPE FCP
ZVM63B: END
```

2. Check the syntax of the change with the **CPSYNTAX** command on the `MAINT 193` disk:

```
===> vmlink maint 193
===> cpsyntax system config f (lpar a09
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
===> cpsyntax system config f (lpar a0a
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

3. Detach the Conversational Monitor System (CMS) file system that contains the SYSTEM CONFIG again:

```
===> vmlink pmaint cf0 < detach >
```

When z/VM IPLs, the EQIDs are created.

Fibre Channel adapters must always be dedicated as pairs that are connected to two fabrics. To dedicate the devices **B802** and **B902** to virtual addresses **FC00** and **FD00**, use the DirMaint `DEDICATE` command as shown in Example 11-2 on page 351.

Example 11-2 Dedicate FCPA as VDEV to a Linux virtual machine

```
===> dirmaint foruser linux2 dedicate FC00 B802
DVHXMT1191I Your DEDICATE request has been sent for processing to DIRMAINT ...
 DVHREQ2288I Your DEDICATE request for LINUX2 at * has been accepted.
 DVHBIU3450I The source for directory entry LINUX2 has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHDRC3451I The next ONLINE will take place via delta object directory.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry LINUX2 have been placed online.
 DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
Ready;
===> dirmaint foruser linux2 dedicate FD00 B902
DVHXMT1191I Your DEDICATE request has been sent for processing to DIRMAINT
...
DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
```

That way, the mapping of the real device that is mapped to FC00 and FD00 in Linux is controlled by z/VM. All of the Linux virtual machines see the virtual adapters FC00 and FD00 only, and they are easier to manage.

To manually dedicate that same pair of FCPs without requiring the virtual machine to log on again, use the following commands:

```
===> ATTACH B802 TO LINUX3 AS FC00
===> ATTACH B902 TO LINUX3 AS FD00
```

As a convention, always keep the range of B8*xx* dedicated as FC00, and B9*xx* as FD00, which simplifies the management of the virtual machines.

After the devices are attached, you can check the WWPN of the adapter with the command:

```
===> QUERY B800 B900
```

These WWPNs (together with the WWPN of the adapters on the storage system) must be configured in their own Fibre Channel zone on the Fibre Channel switch.

11.2.3 Emulated DASD

To simplify the handling of Fibre Channel adapters, you can also define emulated DASD (EDEV), as shown in Figure 11-2. EDEVs are based on Fibre Channel, but they still show up as DASD on z/VM.

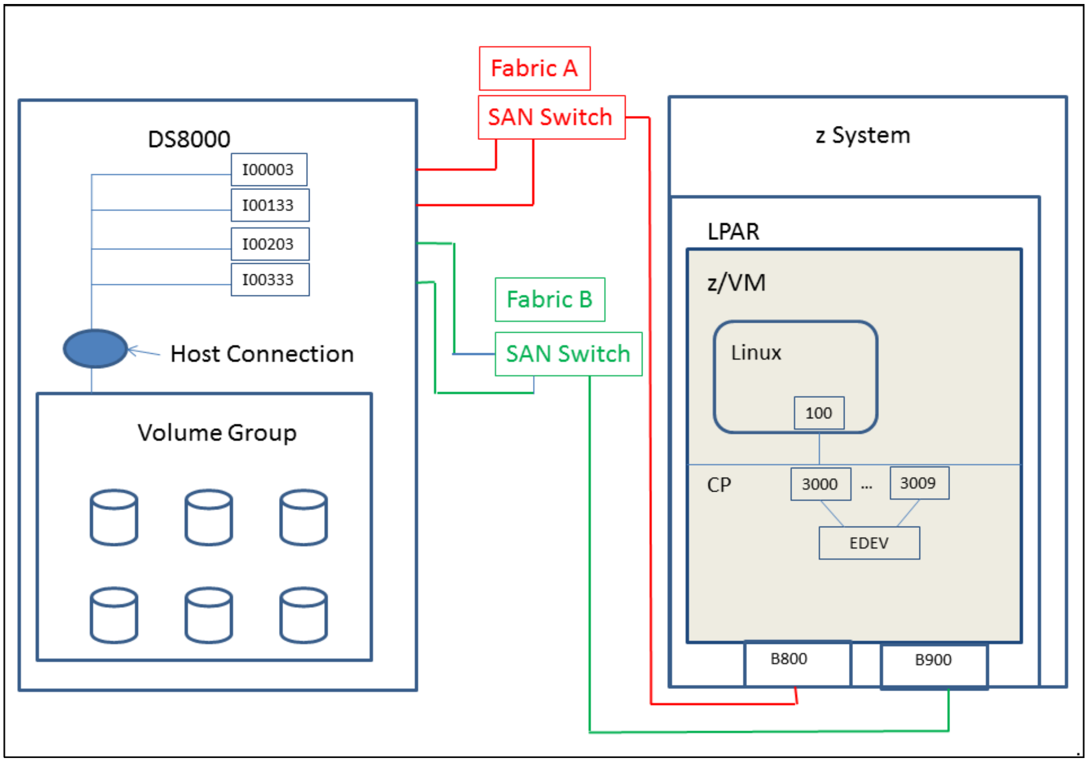


Figure 11-2 EDEV setup overview

Figure 11-2 shows several identifiers in this setup:

- Each of the LUNs in the volume group has a LUN number.
- Each of the I/O devices in the DS8000 has its own WWPN.
- Each of the NPIV adapters on the IBM Z machine has its own WWPN.

The necessary configuration tasks in the SAN are shown:

- On the DS8000, configure all of the LUNs, and put them into a volume group.
- On the z System, get all of the needed NPIV WWPNs from the Support Element (SE). To access them, perform these steps:
  - Log on to the Hardware Management Console (HMC).
  - Select your system. Then, select **Recovery** and **Single Object Operations**.
  - On the SE, select **System Management**, your system, central processor complex (CPC) configuration, and **FCP Configuration**. Look up the NPIV addresses that you want.
- On the DS8000, set up the host connections to the WWPN addresses that you looked up at the SE.
- On the SAN switches, set up a new zone that contains the WWPN addresses of the z System adapters and the WWPN addresses of the DS8000 I/O ports.

After you perform these steps, your environment is prepared to configure the EDEV on z/VM.

Emulated DASD characteristics

The following differences exist between direct-attached FCP and EDEVs:

- EDEVs are based on FCP; therefore, they need a pair of FCP adapters for you to configure this disk type.
- Each of the LUNs that are available to the FCP adapters that are configured for EDEV maps to a single DASD device. Although this setup is only available within z/VM, it is still called *rdev* in that environment.
- Multipathing is configured in z/VM, and the virtual machine systems are not aware of this multipathing.
- The resulting DASDs are fixed-block architecture (FBA) type instead of CKD, and have a default block size of 512 bytes instead of 4k bytes.
- The sizes of EDEVs are determined by the sizes of the volume groups of the storage device only. No correlation exists to 3390 DASD models.
- The IPL of EDEV works like the IPL on the CKD DASD.
- Compared to direct-attached Fibre Channel, the performance is limited.
- EDEV disks are configured as emulated 9336 Model 20. The maximum disk size for CMS disks is 45 GB. The maximum size for FBA disks is 381 GB.
- IBM suggests that clients that are defining disks for use by CMS need to set a practical limit of about 22 GB.

Configuring z/VM to use emulated DASD

Perform the persistent configuration of EDEV in System Configuration (SYSTEM CONFIG). The following example shows how to configure one EDEV with the following components:

- NPIV adapters B800 and B900
- DS8000 I/O ports that are reachable from B800 with WWPNs:
  - 500507630500C74C
  - 50050763050BC74C
- DS8000 I/O ports that are reachable from B900 with WWPNs:
  - 50050763051BC74C
  - 500507630510C74C
- The LUN number is 4010401100000000.
- The rdev of the EDEV is 3000.
- You are using a DS8000 Model 2107.
- The EDEV is on the SSI with the nodes ITSOZVM1 and ITSOZVM2.

Follow this procedure:

1. Log on as MAINT on ITSOZVM2.
2. Run the following commands:

   ===> **`ACCESS 193 G`**

   ===> **`LINK PMAINT CF0 CF0 MR`**

   ===> **`ACCESS CF0 Z`**

   ===> **`XEDIT SYSTEM CONFIG Z`**

3. Page down to the line immediately before the section with Status of Devices.
4. Insert the lines that are shown in Example 11-3 to the SYSTEM CONFIG Z file.

Example 11-3 EDEV SYSTEM CONFIG

```
ITSOZVM1: begin
  EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107,
    FCP_DEV B800 WWPN 500507630500C74C LUN 4010401100000000,
    FCP_DEV B800 WWPN 50050763050BC74C LUN 4010401100000000,
    FCP_DEV B900 WWPN 500507630510C74C LUN 4010401100000000,
    FCP_DEV B900 WWPN 50050763051BC74C LUN 4010401100000000
ITSOZVM1: end
ITSOZVM2: begin
  EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107,
    FCP_DEV B800 WWPN 500507630500C74C LUN 4010401100000000,
    FCP_DEV B800 WWPN 50050763050BC74C LUN 4010401100000000,
    FCP_DEV B900 WWPN 500507630510C74C LUN 4010401100000000,
    FCP_DEV B900 WWPN 50050763051BC74C LUN 4010401100000000
ITSOZVM2: end
```

5. Save and exit XEDIT with the **FILE** command.
6. Run the following commands:

   ===> **CPSYNTAX SYSTEM CONFIG Z (LPAR A09**

   ===> **CPSYNTAX SYSTEM CONFIG Z (LPAR A0A**
7. Check the output of the last two commands for errors, and fix any errors.
8. Run the following command:

   ===> **RELEASE Z (DETACH**

To perform the online configuration without restarting z/VM, complete the following steps:

1. Create a small REXX script that enables the EDEV:

   ===> **XEDIT SETEDEV EXEC A**
2. Insert the lines that are shown in Example 11-4 into the newly created **SETEDEV EXEC A** file.

Example 11-4 Online definition of EDEV devices with SETEDEV EXEC

```
00001 /* REXX */
00002 'SET EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107',
00003    'FCP_DEV B800 WWPN 500507630500C74C LUN 4010401100000000',
00004    'FCP_DEV B800 WWPN 50050763050BC74C LUN 4010401100000000'
```

3. Save that script and run it:

   ===> **SETEDEV**
4. Create another REXX script that enables the additional paths on B900:

   ===> **XEDIT ADDEDEV EXEC A**

5. Insert the lines that are shown in Example 11-5.

Example 11-5 Insert these lines

```
/* REXX */
'SET EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107 ADD PATH',
'FCP_DEV B900 WWPN 500507630510C74C LUN 4010401100000000',
'FCP_DEV B900 WWPN 50050763051BC74C LUN 4010401100000000'
```

6. Repeat all of the previous steps on all of the other nodes of the SSI.
7. Check the paths for the EDEV with the command:

   ===> **QUERY EDEV 3000 DETAILS**

The output of QUERY EDEV 3000 DETAILS looks like output that is shown in Example 11-6.

Example 11-6 Output of QUERY EDEV 3000 DETAILS

```
QUERY EDEV 3000 DETAILS
EDEV 3000 TYPE FBA ATTRIBUTES 2107
  VENDOR: IBM PRODUCT: 2107900 REVISION: 6.30
  BLOCKSIZE: 512 NUMBER OF BLOCKS: 125829120
  PATHS:
    FCP_DEV: B746 WWPN: 5005076309141145 LUN: 4000401F00000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: B746 WWPN: 50050763091B1145 LUN: 4000401F00000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: C746 WWPN: 5005076309149145 LUN: 4000401F00000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: C746 WWPN: 50050763091B9145 LUN: 4000401F00000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
  EQID: 6005076309FFD1450000000000000000F1C600000000077FFFFF
  SERIAL NUMBER: 75KCG71001F
```

During the re-IPL of z/VM, many messages scroll by, including that the original EQID is replaced by an automatic value. This message is expected. It shows that the devices were detected correctly.

All of the configured EDEVs are now ready to use. To use them, handle them as rdev that is available on this z/VM LPAR. The commands to DEDICATE a disk are identical to the commands of configuring CKD DASD.

Cloning FBA EDEV

Run the following commands to clone FBA EDEV 3002 onto 3006:

```
===> attach 3002 to * R/O
DASD 3002 ATTACHED TO MAINT630 3002 R/O
===> attach 3006 to *
DASD 3006 ATTACHED TO MAINT630 3006
Ready;
===> ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
===> sysprint cons
ENTER:
===> input 3002 dasd
ENTER:
```

```
===> output 3006 dasd
ENTER:
===> copy all
HCPDDR711D VOLID READ IS VOL10X
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
===> yes
HCPDDR711D VOLID READ IS ...
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
===> yes
COPYING   VOL10X
COPYING DATA  04/30/15 AT 17.27.46  GMT FROM VOL10X TO ...
INPUT BLOCK EXTENTS         OUTPUT BLOCK EXTENTS
START       STOP             START       STOP
0   20971519                 0   20971519
... // (might take a long time)
END OF COPY
ENTER:
... // (Press Enter.)
END OF JOB
Ready;
```

Managing the EDEV paths for service

The EDEVPATH utility is available “as is” from the z/VM download page. The EDEVPATH utility allows batch control operations to be performed against all of a system’s EDEV paths that share a common trait.

For more information and download instructions, see this web page.

11.2.4 Minidisks

In every z/VM system, minidisks are a widely used method to split full DASD disks into smaller volumes, which are also called *minidisks*. This method can be used with both real CKD DASDs and EDEVs.

Minidisks characteristics

In addition to the smaller size, consider the following important information about minidisks:

- Minidisks can be shared between several z/VM virtual machines inside the same z/VM.
- They are widely used to provide small disks to each Linux virtual machine on the system, which is preferable when every virtual machine must have its own read/write CMS disk.
- DirMaint provides the means to reasonably organize all of the necessary minidisks.
- Minidisks can be used to provide virtual HYPERPAV alias devices that are distributed over the available HYPERPAV alias devices. Therefore, more HYPERPAV alias devices can be set up than are physically configured in the environment.
- When you use HYPERPAV, only full-pack minidisks, including cylinder 0, can be used. To prevent abuse, you can attach the real DASD with the DEVNO statement at the MDISK definition instead of using a volume ID (volid). To further improve system security with this specific type of minidisks, do not enable minidisks for users in SYSTEM CONFIG. Enable minidisks for users only in PROFILE EXEC of AUTOLOG1.
- z/VM provides the means to enable caching on minidisks.
- If no directory maintenance program, such as DirMaint, is used, be careful to not overlap different minidisks.

Configuring z/VM to use minidisks

To add a preformatted disk with LABEL VV1222 to the system, follow this procedure:

1. Add the DASD to the EXTENT CONTROL:

   ===> **`dirmaint dasd add region VV1222 VV1222 3390-03 END START`**

2. Check the result with the command:

   ===> **`dirmaint dasd query region VV1222`**

3. Attach the disk with label VV1222 to the system:

   ===> **`attach 1222 to system`**

4. Ensure that the disk is added to the SYSTEM CONFIG under `User_Volume_List`, as described in step 5 in 6.11.5, "Updating the SYSTEM CONFIG file" on page 192.

Defining MINIDISK for the z/VM virtual machine

When you use DirMaint to manage the system minidisks, ensure that you always add all of the used volumes to the EXTENT POOL. Working around this central configuration can easily result in errors in the future.

To create a list of all available space within the different minidisk regions, run the following command:

===> **`dirmaint FREEXT`**

When you follow the basic label syntax that was introduced, you can see the volumes that are planned for a specific purpose, even in the resulting report. The report is sent to the reader. The procedure to receive the files was explained earlier in this chapter.

The resulting file looks similar to the following example:

```
* * * Top of File * * *
VOLUME   DEVTYPE  -------------- FREE EXTENTS ---------------
VV155B   3390-09  START=      5959 AVAIL=       500
VV155B   3390-09  START=      6959 AVAIL=      3058
VV155F   3390-09  START=      3351 AVAIL=      6666
VV156A   3390-09  START=        14 AVAIL=       189
VV156A   3390-09  START=       855 AVAIL=      9162
VV1222   3390-03  START=         1 AVAIL=      3338
```

Multiple "holes" are within single volumes. An entry for each of these holes is in the report.

To add the volume VV1222 as a full-pack minidisk to the user LNXADMIN on node 2 of the SSI, the volume must be added to the sub-configuration (subconfig) of that user. In our case, we add it with the virtual device number 201:

===> **`dirm for LNXADM-2 AMDisk 201 3390 AUTOV 3338 VV1222 MR`**

Consider the following points about the command:

- AMDISK is used as an abbreviation for "add minidisk".
- The next parameter is the virtual address of the new minidisk.
- 3390 is the device type.
- AUTOV tells DIRMAINT to base on a VOLUME.
- 3338 is the number of cylinders that are used on that volume.
- V1222 is the volume label.
- MR is the link mode for that volume.

Adding a full-pack minidisk from the POOL1 Linux volume pool

Use this command:

```
===> DIRMAINT FOR LINUX3 AMDisk 0702 3390 AUTOG 10016 POOL1 MR
DVHXMT1191I Your AMDISK request has been sent for processing to DIRMAINT
```

This command adds a full-pack minidisk that uses an entire Mod 9 to LINUX3.

11.2.5 HyperPAV

HyperPAV is not a disk device. It is a special device that allows CKD DASD and CKD minidisks to run more than one I/O operation at a time. For disks that are based on FCP, this function is not useful because FCP disks use SCSI command queuing instead.

HyperPAV characteristics

HyperPAV in z/VM for Linux virtual machine can be used in several ways:

- HyperPAV with dedicated DASD

  The Linux system is responsible for managing and serializing I/O requests across the subchannels.

- HyperPAV with minidisks

  The Linux system is not aware of HyperPAV. The Linux subsystem sees the minidisk as a regular DASD and Linux can send only one I/O request at a time to the device. HyperPAV is beneficial only when several minidisks are defined on the same real device or when several virtual machines access the same minidisk at the same time. All of those I/O requests come to z/VM, which handles them and uses HyperPAV aliases, as needed.

- HyperPAV minidisks without operating system exploitation

  A non-exploiting operating system in a virtual machine is either not configured to use HyperPAV or it cannot use HyperPAV. In this case, z/VM can use HyperPAV on behalf of a non-exploiting virtual machine when several virtual machines share the full-pack minidisk by using multiple LINKs.

- HyperPAV minidisks with operating system exploitation

  An exploiting operating system in a virtual machine can control HyperPAV features. This operating system understands how to control and use virtual HyperPAV aliases. Base devices must be defined as full-pack minidisks to the virtual machine. Virtual alias devices are then defined by using the **DEFINE HYPERPAVALIAS** command.

Configuring z/VM to use HyperPAV

Example 11-7 shows how to define HyperPAV to a Linux operating system that uses dedicated DASD. The example dedicates a DASD at virtual device 100 and adds two HyperPAV alias devices from the pool of available HyperPAV alias devices.

Example 11-7 Directory entry for using HyperPAV with dedicated DASD

```
===> dirmaint foruser linux2 dedicate 01FE 15FE
DVHXMT1191I Your DEDICATE request has been sent for processing to DIRMAINT ...
 DVHREQ2288I Your DEDICATE request for LINUX2 at * has been accepted.
 DVHBIU3450I The source for directory entry LINUX2 has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHDRC3451I The next ONLINE will take place via delta object directory.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry LINUX2 have been placed online.
 DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
```

```
Ready;
===> dirmaint foruser linux2 dedicate 01FF 15FF
DVHXMT1191I Your DEDICATE request has been sent for processing to DIRMAINT
...
DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
Ready;
===> dirmaint foruser linux2 dedicate 0100 1570
DVHXMT1191I Your DEDICATE request has been sent for processing to DIRMAINT
...
DVHREQ2289I Your DEDICATE request for LINUX2 at * has completed; with RC = 0.
```

After you dedicate a real HyperPAV alias to a single virtual machine, this specific alias device cannot be shared within this z/VM instance anymore.

Using shared HyperPAV for minidisks

The following example defines a full-pack minidisk at virtual device `102` and six virtual HyperPAV aliases at virtual devices `1FA-1FF`:

```
USER LINUX3 LNX4VM 768M 1G G
   INCLUDE LNXPDFLT
   COMMAND DEFINE HYPERPAVALIAS 1FA FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 1FB FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 1FC FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 1FD FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 1FE FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 1FF FOR BASE 102
   OPTION APPLMON
   MDISK 0100 3390 0001 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0101 3390 5009 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0102 3390 DEVNO 1368 MR LNX4VM LNX4VM LNX4VM
```

When the virtual machine is logged on, the disks are defined:

```
...
00: NIC 0AD0 is created; devices 0AD0-0AD2 defined
00: NIC 0AD0 is connected to VSWITCH SYSTEM VSW1
00: DASD 01FA DEFINED
00: DASD 01FB DEFINED
00: DASD 01FC DEFINED
00: DASD 01FD DEFINED
00: DASD 01FE DEFINED
00: DASD 01FF DEFINED
...

# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   3521MB    901440
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440
```

Follow the procedure for adding new disks according to your distribution (edit `/etc/dasd.conf` for RHEL (or use the **`dasd_configure`** command in SLES) as described in 13.1, “Adding disk space to Linux virtual machines” on page 398.

After devices 102, 200 - 205 are configured, the output of the **lsdasd** command changes:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.01FA alias                        ECKD
0.0.01FB alias                        ECKD
0.0.01FC alias                        ECKD
0.0.01FD alias                        ECKD
0.0.01FE alias                        ECKD
0.0.01FF alias                        ECKD
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0300   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0301   active      dasdc     94:8    FBA   512    512MB     1048576
0.0.0102   active      dasdd     94:12   ECKD  4096   7043MB    1803060
```

No other configuration is needed. From now on, whenever /dev/dasdd is used, Linux uses the base device and alias devices to distribute the workload. No multipathing is needed for HyperPAV to work in an exploiting Linux.

If another device, which comes from the same logical control unit as device 102 in the previous example, is added, it also uses the same virtual HyperPAV devices. If a real device from a different logical control unit is added, it needs a new set of virtual HyperPAV aliases to be added.

Consider the following information when you define virtual HyperPAV aliases:

- The base device must be defined as a full-pack minidisk, including cylinder 0.
- Because the base device is defined as a full-pack minidisk, Linux has control of cylinder 0, also. Therefore, **dasdfmt** overwrites the volume serial with 0xyyyy where yyyy is the virtual device number. To solve this issue, use minidisk assignment by device number.
- The number of virtual HyperPAV aliases that is defined in one virtual machine for one logical control unit cannot be higher that the number of real aliases in that logical control unit's pool of aliases. It does not matter whether all virtual HyperPAV aliases in the virtual machine are defined to one base device or whether they are spread among several devices, they all act as one pool of aliases for a specific logical control unit.

  For example, assume that a logical control unit exists with 20 real devices and 20 aliases in a HyperPAV pool. We want to define four Linux images (each image with five real devices). To configure each Linux for the maximum throughput, we define each Linux with 20 virtual HyperPAV aliases. They can be defined to one real device or spread among all real devices because the results are the same. A total of 20 aliases exist for five devices in a virtual machine:

  ```
  USER LINUX3...
  ...
  COMMAND DEFINE HYPERPAVALIAS 1EC FOR BASE 100
  ...
  COMMAND DEFINE HYPERPAVALIAS 1FF FOR BASE 100
  MDISK 100 3390 0 END VOL001 MR
  ...
  MDISK 104 3390 0 END VOL005 MR
  ```

  Although you achieve the same 20 aliases for five devices with the following definition, the former output is easier to read:

  ```
  USER LINUX6...
  ...
  ```

```
COMMAND DEFINE HYPERPAVALIAS 1EC FOR BASE 100
COMMAND DEFINE HYPERPAVALIAS 1ED FOR BASE 100
...
COMMAND DEFINE HYPERPAVALIAS 1EF FOR BASE 101
COMMAND DEFINE HYPERPAVALIAS 1F0 FOR BASE 101
...
COMMAND DEFINE HYPERPAVALIAS 1F1 FOR BASE 102
COMMAND DEFINE HYPERPAVALIAS 1F2 FOR BASE 102
...
COMMAND DEFINE HYPERPAVALIAS 1F4 FOR BASE 104
MDISK 100 3390 0 END VOL001 MR
...
MDISK 104 3390 0 END VOL005 MR
```

For more information about HyperPAV in z/VM, see *z/VM CP Planning and Administration*, SC24-6178.

11.3 Network attachment options and considerations

For more information about connectivity, see Chapter 2, “Planning” on page 33. On IBM Z and LinuxONE systems, different networking methods are available, which are provided by the I/O subsystem that controls the LPARs, such as IBM PR/SM, or by the z/VM hypervisor.

11.3.1 z/VM virtual switch (VSWITCH)

No special action is necessary to use the VSWITCH interfaces that are at the address triplet 0AD0, 0AD1, and 0AD2. The groundwork for the use of the VSWITCH interfaces was set up and configured as part of the shared user profiles.

11.3.2 Direct-attached Open Systems Adapter

Linux uses the identical drivers to run a direct-attached Open Systems Adapter (OSA) that it uses to run VSWITCH interfaces. With a direct-attached OSA, you tradeoff lower TCO, high availability, and stall prevention for a possible faster network connection to the external network.

Important: The direct-attached OSA configuration is not a recommended configuration. Use it with extreme caution. It is a single point of failure in an otherwise highly available environment. It can also become packed and stall, which triggers rollbacks or failures in fault-intolerant consumers. The VSWITCH does not exhibit this weakness and remains the recommended method.

As 10-gigabit Ethernet (10 GiE) becomes more prevalent, the small number of possible use cases for this scenario dwindled even further. An exceptionally small number of cases exist in which a direct-attached OSA is necessary when the VSWITCH uses 10 GiE uplinks.

11.3.3 Configuring z/VM to provide direct-attached OSA interfaces

The configuration that is needed to provide OSA interfaces to a virtual machine is basically the same as dedicating three suitable OSA device addresses to the virtual machine.

Example 11-8 shows a sample entry in the user directory for the triple 2104, 2105, and 2103 that maps to 0AD0, 0AD1, and 0AD2. This mapping results in the physical OSA ports 2104-2013 being dedicated specifically to the user LINUX3 and appearing as virtual OSA ports 0AD0-0AD3.

Example 11-8 Direct-attached OSA

```
USER LINUX3 LNX4VM 768M 1G G
...
   DEDICATE 0AD0 2104
   DEDICATE 0AD1 2105
   DEDICATE 0AD2 2103
```

This mapping also can be accomplished by using the following DirMaint commands:

```
DIRMAINT FOR LINUX3 DEDICATE 0AD0 2104
DIRMAINT FOR LINUX3 DEDICATE 0AD1 2105
DIRMAINT FOR LINUX3 DEDICATE 0AD2 2103
```

11.3.4 Configuring z/VM to provide HiperSockets network interfaces

The rules for numbering devices for HiperSockets are the same as the rules for numbering devices for OSA. However, ensure that each triplet is used only one time within a CSS.

Example 11-9 shows an entry in the user directory for the triple 5500, 5501, and 5502 that maps to only the same device addresses.

Example 11-9 User directory entry for numbering devices for HiperSockets

```
USER LINUX3 LNX4VM 768M 1G G
...
   DEDICATE 5500 5500
   DEDICATE 5501 5501
   DEDICATE 5502 5502
```

11.4 Common DirMaint tasks

When the VMSSI feature is enabled, the management of the systems without an automated directory management product becomes a highly complex task. Therefore, we assume that you are using the IBM DirMaint directory management product. Other software products might provide similar functions.

You must perform numerous ongoing tasks for your z/VM systems, such as adding minidisks, working with directory entries, and modifying parameters. The details that are needed to accomplish many of the common tasks are provided. Also, tips are provided to facilitate these tasks.

11.4.1 DirMaint and the user directory characteristics

The user directory controls the resources of all of the virtual machines within a specific z/VM. For several nodes in an SSI, several instances of the user directory are needed. DirMaint helps to keep all of the necessary resource information synchronized across all of the SSI. DirMaint offers these important features:

- DirMaint keeps changes in sync across all of the SSI.
- It handles all minidisk assignments and prevents user errors in minidisk assignments.
- DirMaint supports all user directory statements within z/VM.
- The sequence of directory statements is handled automatically.

Several standard tasks need to be performed frequently on z/VM. We provide assistance for these common tasks.

11.4.2 Checking the status of DirMaint and subcomponents

The DirMaint QUERY command opens the query window (Figure 11-3), where you can complete various fields to obtain information about the status. Place an `X` in any one of the options and press PF5 to submit for processing. Use the following command:

```
===> dirmaint query
```

```
--------------------------------DirMaint QUERY-------------------------------

 Request current system information from the DIRMAINT server.

 Select or complete one of the following:
   - To show the DirMaint system level file:
       _ DVHLEVEL
   - To show the Work Unit Control File queue:
       _ UNASSIGNED
   - To show a specific Work Unit Control File:
        WORKUNIT ===>
   - To show the status of one or all DATAMOVE machines:
        DATAMOVE  Userid ===>            or   _ * (ALL)
     To additionally show current pending elements status for these machines:
       _ RETRY (optional)
```

Figure 11-3 DirMaint QUERY window

11.4.3 Adding a USER to z/VM by using a prototype

A USER entry in the z/VM user directory is also known as a *Single Configuration virtual machine* (SCVM). It needs to be prepared to be relocated within the SSI cluster. Normally, a USER can be run on any one member at a time anywhere in the cluster.

Typically, users are added into z/VM by using PROTOTYPEs when DirMaint is in use. The creation of prototypes was explained in section 6.13.2, “Role-based access controls and CP privilege classes” on page 206.

Using the LNXPROTO prototype and optionally CLONEDISK

Run this command to use the LNXPROTO prototype. Running CLONEDISK is optional.

```
===> dirmaint add linux5 like lnxproto pw lnx4vm
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT ...
Ready;
  DVHREQ2288I Your ADD request for LINUX5 at * has been accepted.
  DVHBIU3450I The source for directory entry LINUX5 has been updated.
  DVHBIU3424I The next ONLINE will take place immediately.
  DVHDRC3451I The next ONLINE will take place via delta object directory.
  DVHRLA3891I Your DSATCTL request has been relayed for processing.
  DVHBIU3428I Changes made to directory entry LINUX5 have been placed online.
  DVHSCU3541I Work unit 30164323 has been built and queued for processing.
  DVHSHN3541I Processing work unit 30164323 as MAINT630 from ENDVM363,
  DVHSHN3541I notifying MAINT630 at ENDVM363, request 394.1 for LINUX5 SSI
  DVHSHN3541I node *; to: AMDISK 0100 3390 AUTOG 10016 POOL1 MR
  DVHBIU3450I The source for directory entry LINUX5 has been updated.
  DVHBIU3424I The next ONLINE will take place immediately.
  DVHDRC3451I The next ONLINE will take place via delta object directory.
  DVHRLA3891I Your DSATCTL request has been relayed for processing.
  DVHBIU3428I Changes made to directory entry LINUX5 have been placed online.
  DVHSHN3430I AMDISK operation for LINUX5 address 0100 has finished
  DVHSHN3430I (WUCF 30164323).
  DVHREQ2289I Your ADD request for LINUX5 at * has completed; with RC = 0.
```

Wait a few minutes for asynchronous processing to complete, then proceed with the **clonedisk** operation, if you want. For this example to work correctly, both the source and target IDs, LINUX5 and LINUX3 in this case, must be logged off.

```
===> dirmaint for linux5 clonedisk 0100 linux3 0100
DVHXMT1191I Your CLONEDISK request has been sent for processing to DIRMAINT ...
Ready;
DVHREQ2288I Your CLONEDISK request for LINUX5 at * has been accepted.
 DVHSCU3541I Work unit 30183646 has been built and queued for processing.
 DVHSHN3541I Processing work unit 30183646 as MAINT630 from ENDVM363,
 DVHSHN3541I notifying MAINT630 at ENDVM363, request 453 for LINUX5 SSI
 DVHSHN3541I node *; to: CLONEDISK 0100 LINUX3 0100
 DVHBIU3450I The source for directory entry DATAMOVE has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
...
  DVHBIU3428I Changes made to directory entry LINUX5 have been placed online.
  DVHREQ2289I Your CLONEDISK request for LINUX5 at * has completed; with RC = 0.
DVHSHN3430I CLONEDISK operation for LINUX5 address 0100 has finished ...
```

After this command completes, remember to enroll the user in the file pool and generate the ALIASES, as described in section 6.16.5, "Enrolling the Linux virtual machines as USERS" on page 243.

11.4.4 Adding a user to z/VM without the use of a prototype

Complete the following steps to add a user to z/VM without the use of a prototype:

1. Log on as MAINT or MAINT630.
2. Create a file <USERID> DIRECT A:

```
===> xedit LINUX1 DIRECT A
USER LINUX1 LNX4VM 1G 2G G

   INCLUDE LNXPDFLT
```

3. Add the user to the directory:

```
===> dirmaint add LINUX1
```

4. After this process completes, erase the temporary work file:

```
===> erase LINUX1 DIRECT A
```

5. Enroll the user in the file pool and generate the ALIASES, as described in 6.16.5, "Enrolling the Linux virtual machines as USERS" on page 243.

11.4.5 Adding an IDENTITY to z/VM by using a prototype

For more information about this process, see 6.17, "Creating identity LNXADMIN for Linux administration" on page 246.

11.4.6 Adding an IDENTITY to z/VM without the use of prototypes

An IDENTITY entry in the z/VM user directory is also known as a *multi-configuration virtual machine* (MCVM). An IDENTITY entry is configured so that the resources that are defined to it are unique to each member node. An identity is not eligible for relocation, which makes it possible to have multiple concurrent logins to the same identity on different member nodes.

Complete the following steps:

1. Log on as MAINT or MAINT630.
2. Create a temporary directory entry file. In this example, we are adding LNXADMIN, a privileged virtual machine with elevated rights to user classes B, D, and E, in addition to the usual class G for General:

```
==> xedit LNXADMIN DIRECT A
IDENTITY LNXADMIN LNX4VM 768M 2G BDEG

   INCLUDE LNXPDFLT
   OPTION LNKNOPAS
Create the needed subconfig definitions:

==> xedit LNXADM-1 DIRECT
SUBCONFIG LNXADM-1
   MDISK 0100 3390 1 10016 VM1567 MR LNX4VM LNX4VM LNX4VM
   MDISK 0200 3390 1 10016 VM1568 MR LNX4VM LNX4VM LNX4VM

===> xedit LNXADM-2 DIRECT
SUBCONFIG LNXADM-2
   MDISK 0100 3390 1 10016 VM1569 MR LNX4VM LNX4VM LNX4VM
   MDISK 0200 3390 1 10016 VM156A MR LNX4VM LNX4VM LNX4VM
```

3. Add the definitions to the directory:

```
===> dirm add LNXADMIN
===> dirm add LNXADM-1 build on ITSOVM1 in LNXADMIN
===> dirm add LNXADM-2 build on ITSOVM2 in LNXADMIN
```

4. Erase the temporary work files:

```
===> erase * direct a
```

11.4.7 Changing the amount of memory that is assigned to a user

Two different values are in the user definition. The value for STORAGE defines how much main memory the virtual machine gets at log-on time. The value for MAXSTORAGE defines how much memory can be used by issuing **define storage**.

Consider the following examples:

- Retrieve information about the current memory setting of LNXADMIN:

```
===> dirm for LNXADMIN STORAGE ?
```

- Set the default storage size for LNXADMIN to 800 MB:

```
===> dirm for LNXADMIN STORAGE 800M
```

- Set the maximum storage size for LNXADMIN to 2 GB:

```
===> dirm for LNXADMIN MAXSTORAGE 2G
```

11.4.8 Modifying a user

Complete the following steps to manually change a profile:

1. Get the entry from the directory:

```
===> dirm for LNXPDFLT get
DVHXMT1191I Your GET request has been sent for processing to DIRMAINT at
DVHXMT1191I ITSOZVM1 via DIRMSAT2.
Ready; T=0.01/0.01 10:39:28
 DVHREQ2288I Your GET request for LNXPDFLT at * has
 DVHREQ2288I been accepted.
 DVHGET3304I Directory entry  LNXPDFLT is now locked.
 DVHREQ2289I Your GET request for LNXPDFLT at * has
 DVHREQ2289I completed; with RC = 0.
RDR FILE 0081 SENT FROM DIRMAINT PUN WAS 0730 RECS 0024 CPY  001 A NOHOLD
NOKEEP
```

2. Receive the file from the reader:

```
===> receive 81 (replace
```

3. Edit the entry:

```
===> xedit LNXPDFLT direct a
```

4. Send the changed definition back to the directory:

```
===> dirm for LNXPDFLT replace
```

5. Erase the local copy of the definition. It contains passwords, and DirMaint is the authoritative source; therefore, you do not need to retain the temporary copy:

```
===> erase LNXPDFLT direct A
```

11.4.9 Deleting a user

Sometimes, it is necessary to remove a user from the directory. You can erase a user with a single command:

```
===> dirm for <userid> purge
```

If you want to also ensure that the abandoned minidisk extents are scrubbed clean, use this command instead:

```
===> dirmaint for <userid> purge (clean
```

11.4.10 Adding a minidisk to a user or identity

The setup procedure for this task is described in “Configuring z/VM to use minidisks” on page 357.

11.4.11 Getting a copy of the user directory

Complete the following steps to get a complete directory definition:

1. Ask DirMaint to send the user directory:

   ```
   ===> dirm user withpass
   ```

2. Receive the file:

   ```
   ===> receive <nr on reader> (repl
   ```

11.4.12 Getting and updating the EXTENT CONTROL file

The EXTENT CONTROL file holds all information about the disks that are available for minidisk storage. You can modify the contents with the command **dirm dasd**, but you can also get the complete file and replace it with changes.

Complete the following steps:

1. Get the EXTENT CONTROL from the directory:

   ```
   ===> dirm send extent control
   ```

2. Receive the file from the reader:

   ```
   ===> receive <nr on reader> (repl
   ```

3. Edit the file:

   ```
   ===> xedit extent control
   ```

4. Send the file back:

   ```
   ===> dirm file extent control
   ```

5. Reload that file in the system:

   ```
   ===> dirm rldext
   ```

11.4.13 Cleaning up the work units

At times, the DirMaint commands never finish and leave work units behind. In general, this situation rarely occurs.

Before you cancel a work unit that appears to be stuck, use the DirMaint commands to reload the code and data first and allow several minutes for them to complete. If this action does not resolve the issue, complete the following steps:

1. Check for active work units:

```
===> dirm status workunit all
DVHXMT1191I Your STATUS request has been sent for processing to DIRMAINT
DVHXMT1191I at ITSOZVM1 via DIRMSAT2.
Ready; T=0.01/0.01 11:01:46
 DVHREQ2288I Your STATUS request for MAINT at *
 DVHREQ2288I has been accepted.
 DVHSTT3419I The following active Work Unit
 DVHSTT3419I Control Files currently exist:
 DVHSTT3419I 13153849
 DVHREQ2289I Your STATUS request for MAINT at *
 DVHREQ2289I has completed; with RC = 0.
```

2. Display more information about the work unit in question:

```
===> dirm status workunit 13153849
DVHXMT1191I Your STATUS request has been sent for processing to DIRMAINT
DVHXMT1191I at ITSOZVM1 via DIRMSAT2.
Ready; T=0.01/0.01 11:02:11
 DVHREQ2288I Your STATUS request for MAINT at *
 DVHREQ2288I has been accepted.
....
 DVHSTT3419I 13153849 was created by the command:
 DVHSTT3419I AMDISK 0200 3390 AUTOV 500 VV1560 MR
 DVHSTT3419I LABEL DAT100
....
 DVHSTT3419I NTRIED UNLOCK 0200 DATAMOV2 NOMSG
 DVHREQ2289I Your STATUS request for MAINT at *
 DVHREQ2289I has completed; with RC = 0.
```

3. Cancel and roll back the specified work unit:

```
===> dirm workunit 13153849 cancel
```

4. Clean up the datamove:

```
===> dirm for datamove cleanup cancel
```

11.4.14 Checking the DirMaint disk map

To check for overlaps or holes on DASD, read the disk map of the user directory. To request the disk map, use the following command:

```
===> dirmaint dirmap
```

Then, receive the file, or use peek to look at the results.

11.4.15 Dedicating crypto domains

When you use hardware crypto engines, it is important to understand that certain functions are only available when a crypto domain is dedicated to the virtual machine. On z13, a number of patches are required for z/VM to make this type of hardware available.

To check for currently available crypto domains, run the command:

```
===> query crypto ap
q crypto ap
AP 000 CEX5C  Domain 005 available    free                  unspecified
AP 000 CEX5C  Domain 006 available    free                  unspecified
AP 000 CEX5C  Domain 007 available    free                  unspecified
AP 002 CEX5C  Domain 005 available    free                  unspecified
AP 002 CEX5C  Domain 006 available    free                  unspecified
AP 002 CEX5C  Domain 007 available    free                  unspecified
...
Ready; T=0.01/0.01 14:54:17
                                                            RUNNING   ITSOZVM2
```

To dedicate domain 5 from AP 0 and 2, run the following command:

```
===> dirm for linux4 crypto domain 6 apded 0 2
```

To check the currently assigned domains for a virtual machine, run the following command:

```
===> dirm for linux4 crypto ?
DVHXMT1191I Your CRYPTO request has been sent for processing to DIRMAINT
DVHXMT1191I at ITSOZVM1 via DIRMSAT2.
Ready; T=0.01/0.01 14:57:26
 DVHREQ2288I Your CRYPTO request for LINUX4 at * has
 DVHREQ2288I been accepted.
 DVHCRT3337I The current CRYPTO statement is as
 DVHCRT3337I follows in the LINUX4 directory entry:
 DVHCRT3337I CRYPTO DOMAIN 6 APDEDICATED 0 2
```

12

Monitoring z/VM and Linux

This chapter describes how to monitor z/VM and Linux.

For more information about z/VM performance and monitoring, see Chapter 11, "Monitoring performance and capacity", in *Getting Started with Linux on System z*, SC24-6096.

Many z/VM monitoring tools, such as CA VM:Monitor, IBM z/VM Performance Toolkit, IBM Tivoli OMEGAMON® XE for z/VM and Linux, and products from IBM Velocity Software, are available. IBM z/VM Performance Toolkit also is described in this chapter.

This chapter includes the following topics:

12.1 Using basic z/VM commands

z/VM features many commands to monitor the state of the system. **CP INDICATE** is the most commonly used, but other commands can also monitor the state of the system. For more information, see this web page.

12.1.1 Using the INDICATE command

z/VM features several basic commands, such as **INDICATE**. Many **INDICATE** parameters can be included as command-line options. Use the **HELP INDICATE** command for a basic understanding and then press F11 for help about each parameter.

INDICATE LOAD

If no parameter is specified, **INDICATE LOAD** is the default option. Two versions are available, depending on whether the issuing virtual machine has privilege class G or class E. Class G users can use the **INDICATE** command to display recent contention for system resources, environment characteristics, and measurements of resources that are used by their virtual machine.

The output from virtual machines with class E privilege (for example, MAINT or OPERATOR) is shown. The lines are numbered for clarity with a description that follows the output:

```
===> ind load
 1 AVGPROC-000% 04
 2 MDC READS-000068/SEC WRITES-000001/SEC HIT RATIO-099%
 3 PAGING-0/SE
 4 Q0-00001(00000)                      DORMANT-00012
 5 Q1-00000(00000)           E1-00000(00000)
 6 Q2-00001(00000) EXPAN-001 E2-00000(00000)
 7 Q3-00001(00000) EXPAN-001 E3-00000(00000)
 8
 9 PROC 0000-000% CP   VM     PROC 0001-000% CP   VL
10 PROC 0002-000% IFL  VM     PROC 0003-000% IFL  VL
11
12 LIMITED-00000
```

The **INDICATE LOAD** command gives a snapshot of current system performance. Except for the counts of virtual machines in various queues and the limited list, the values that you see here are a smoothed average over the past 4 minutes. z/VM performance analysts tend to focus on the following areas:

- AVGPROC on line 1 gives the overall processor utilization, which is 38% in this example. The number that follows it is the number of online processors, 04 in this example. The individual processor utilization is shown on lines 9 and 10. Glance at these numbers to see whether they are balanced. An imbalance is acceptable in certain situations, such as low utilization scenarios or cases where enough users are not ready to run virtual processors to keep the physical processors busy.

  One of the processors is a Master processor. All of the other processors are Alternate processors. Imbalance can result from performing these functions. Another imbalance comes from vertical CPU management.

- Minidisk cache (MDC) statistics are provided on the second line. The effectiveness of MDC can be judged by the combination of the READS rate and the HIT RATIO. If both are high, many physical I/Os are avoided due to the MDC feature.

  For a system with a high I/O rate, which is composed of reads plus writes, a high proportion of reads, and a good hit ratio for those reads (90% or greater), the real, physical I/O avoidance can be high. Avoidance can be as high as 50% in certain cases. However, do not assume that a high HIT RATIO with a low value for the reads rate is good. (A 100% hit ratio with only 1 I/O per second is meaningless.)

- Line **3** describes more storage (memory) management. The PAGING rate is important. Higher values often affect performance. PAGING can be at least partially offset by increasing the number of page volumes, but a more thorough examination of this problem is advisable whenever it arises.

- On lines **4** - **7**, you see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into three classes (1 - 3) and a special extra class that is labeled *zero*. So the column of Q_x values and E_x represent the virtual machines in the dispatch list and the eligible list.

  The most important value to validate is to ensure that no virtual machines are in the Eligible list: E1, E2, or E3, which implies that z/VM stopped dispatching virtual machines to avoid overcommitting resources. This system requires further investigation that might lead to tuning or hardware addition in extreme cases. Do not worry about the values in parentheses.

INDICATE QUEUES EXP

Another useful command to help you understand the state of the system is the **INDICATE QUEUES EXP** command, for example:

```
===> ind q exp
MAINT          Q1 R00   00001623/00001552   .I..  .0004
TCPIP          Q0 PS    00003496/00003178   .I..  99999
```

This class E command displays the virtual processors that are associated with a specific virtual machine (that can have multiple virtual processors), the queue (dispatch list, eligible list, or limit list) that they are in, and their states. This view is a snapshot in time. Again, you want to check this output to ensure that no virtual machines are in the eligible list. The normal virtual processors in the dispatch list are Q_x (x=1, 2, or 3). The eligible list is marked as E_x.

The third column in the example also gives the state of the virtual processor, which can be helpful to get an idea of how the virtual processors might be constrained. Virtual processors that are running in the snapshot period are marked with an *RNN* where *NN* is the processor number they are on. An R without a number means that the virtual processor is ready to run, but no processor is available.

> **Note:** The virtual machine that issues the **INDICATE** command will always be one of the running machines.

Other states are documented in the help for **IND Q EXP**. You do not need to be concerned about the other columns unless detailed analysis is required or if IBM support requests it. Also, always remember that this output is only a snapshot in time. Repeating this command over time gives you a more accurate picture of your z/VM system. A single snapshot cannot be regarded as indicative.

CP INDICATE ACTIVE

Use **INDICATE ACTIVE** to display:

- The total amount of active users in a specific time
- The number of users in the dispatch, eligible, and dormant lists that were active in a specific period

> **Note:** Example 12-1 demonstrates the use of the **INDICATE ACTIVE** command and its results.

CP INDICATE QUEUES

Use the **INDICATE QUEUES** command to display the user IDs and their associated dispatching queue. If users have a virtual multiprocessor, you might see more than one entry of a single user.

> **Note:** The output of the **CP INDICATE QUEUES** command provides the following information:
>
> - List of virtual machines in priority order.
> - If this command frequently shows users in the E1, E2, and E3 list, you likely have a constraint in your system resources.
> - The third column explains why a virtual machine is in a wait state:
>   - Rnn: Current RUNUSER on the specified real processor, where *nn* is the processor ID
>   - IO: Waiting for I/O
>   - PS: PSW wait (enabled wait state)
>   - PG: Waiting for paging

Example 12-1 INDICATE ACTIVE command and its results

```
cp ind queues
TCPIP         Q0 PS    00004441/00004070
MAINT720      Q1 R00   00000548/00000542
```

CP INDICATE I/O

Use the **INDICATE I/O** command to identify the virtual machines that are in an I/O wait state and the real I/O device number on which they are waiting. Repeat this command several times to see the pattern.

> **Note:** Consider the following points:
>
> - **INDICATE I/O** shows virtual machines that are in an I/O wait start and 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 issue might indicate that too many minidisks are on the same DASD or you have a DASD controller bottleneck in which too many DASD are performing I/O on the same controller.

CP INDICATE USER

Use the **INDICATE USER** command to display the resources that are used or occupied by a virtual machine or system. CP displays a set of statistics for each of virtual machine. If you do not include any user ID after this command, the default is the user ID where you are issuing the command.

> **Note:** This command shows the virtual machine definition as USERID, STOR, and other characteristics. It also shows the amount of pages in real and expanded storage. If you want to see whether this user ID is running, run this command several times to see the amount of resources changing or if the number of I/Os is increasing.

CP INDICATE PAGING

Run the **INDICATE PAGING** command to display:

- A list of the virtual machines in page wait status
- The number of pages this user ID has in the expanded storage and auxiliary storage (DASD)

12.1.2 CP Query commands

The CP Query commands can be used to display information about the system and users. Another set of z/VM commands helps to get more information about the z/VM system environment and what was set to a specific virtual machine.

cp query srm

The **cp query srm** command (see Example 12-2) displays system-wide parameters that are used by the scheduler to set the priority of system resource access. These parameters define the size of the time slice and the access to the resources for different user classes as seen by the scheduler. Each of these parameters has a default setting that is suitable for most environments.

The **CP SET SRM** command is useful if you must control users' use of resources that depend on the class of user. Use this command carefully and monitor usage because eligible lists can occur if not used properly.

The System Resource Manager (srm) parameters that are contained in this display includes the following information:

- `iabias`: Interactive bias, used to favor short transactions
- `ldubuf`: Controls z/VM's tolerance for guests that induce paging
- `storbuf`: Can be used to encourage z/VM to overcommit main memory
- `dispatching minor timeslice`: Controls the length of time that a single user holds onto a processor

Example 12-2 cp query srm command and results

```
q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=300% Q2=250% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
DISPATCHING MINOR TIMESLICE = 5 MS
MAXWSS : LIMIT=9999%
```

```
...... : PAGES=396235760
XSTORE : ---
LIMITHARD METHOD: CONSUMPTION
POLARIZATION: VERTICAL
GLOBAL PERFORMANCE DATA: ON
EXCESSUSE: CP-MEDIUM  ZAAP-MEDIUM  IFL-MEDIUM  ICF-MEDIUM  ZIIP-MEDIUM
CPUPAD: CP-100%  ZAAP-100%  IFL-100%  ICF-100%  ZIIP-100%
DSPWDMETHOD: RESHUFFLE
UNPARKING: MEDIUM
```

CP QUERY ALLOC

Use **QUERY ALLOC** to display the number of cylinders or pages that are allocated, in use, and available for DASD volumes that are attached to the system. Use **Q ALLOC PAGE** for detailed information about paging space or **Q ALLOC SPOOL** for detailed information about spool space. You can define it dynamically, but if the system restarts, this definition is lost. For the definition to be permanently set up, you must define on SYSTEM CONFIG file, in the PARM disk of z/VM.

The **CP QUERY ALLOC** command:

- Displays a list of the paging DASDs and the allocation amount at that specific moment.
- Shows a summary of the total amount usage.

Note: Do not mix different types of DASD or different models of DASD because doing so can affect the algorithms and the system performance.

CP Q SHARE

Use the **Q SHARE** ‘userid’ to display the given percentage of the system available resources can use. This percentage can be set as **ABSOLUTE** or **RELATIVE** percentage. As default, the **RELATIVE** percentage of each user ID is 100.

Note: A virtual machine receives its portion of any resource (processors, real storage, and so on) according to its SHARE setting.

CP QUERY QUICKDSP

Use the **Q QUICKDSP** ‘userid’ to display whether this option is ON or OFF for that specific user ID. If it is ON, that user ID is be moved in the eligible list so it can be dispatched without waiting on the queues.

Other query commands that might be useful when investigating performance issues are listed in Table 12-1.

Table 12-1 CP QUERY commands

Query	Description
CACHE	Use QUERY CACHE to display caching status for all storage subsystems that support caching when investigating I/O-related problems.
CHPIDS	Use QUERY CHPIDS to display all 256 of the machine’s channel paths and their physical status. I/O performance problems can occur if CHPIDs are offline or not available.
CPLOAD	Use QUERY CPLOAD to display information regarding the last CP IPL. The information that is displayed includes the location of the CP module that was last used, the location of the parm disk, and how CP was started.

Query	Description
CPOWNED	Use QUERY CPOWNED to display the list of CP-owned DASD volumes. If paging or spooling problems occur, this list can be checked to ensure that all required volumes are available.
FRAMES	Use QUERY FRAMES to display the status of host real storage. It might be necessary to use this command if you are getting users in the eligible list when you need to check storage utilization.
MDC	Use QUERY MDC from a Class B user to: ► Query minidisk cache settings for the entire system, for a real device, an active minidisk, or a minidisk defined in the directory ► Query a user's ability to insert data into the cache Useful if I/O problems are being investigated.
PATHS	Use QUERY PATHS to display: ► All paths installed to a specific device or range of devices ► Installed path status Use when investigating I/O problems.
PENDING	Use QUERY PENDING command to display the device commands that you entered and optionally, that others entered for which the associated asynchronous function is not yet completed. Can be useful when investigating hung users or I/O problems.
QIOASSIST	Use QUERY QIOASSIST to determine the status of the queue-I/O assist for a virtual machine. Might be used when investigating I/O or networking problems.
RESERVED	Use QUERY RESERVED to display the number of reserved real storage frames. Use SET RESERVED to reserve pages of storage for a user. It is useful if tuning users in storage resulted in constraining the system.
SRM	User QUERY SRM to display the settings of the System Resource Manager, including IABIAS, LDUBUF, STORBUF, and so on.
STOR	Use QUERY STORAGE or QUERY STORE to display the size of real storage.
SYSTEM	Use QUERY SYSTEM to display current user access to a system DASD volume.

CP Set commands

Several CP Set commands are available that can be used to change some performance characteristics of the entire system or of a single user.

cp set share command

The **`cp set share`** command changes the system-resource-access priority for users. Two parameters are available (ABSOLUTE and RELATIVE):

- `ABSolute nnn%` : Specifies that this user is to receive a target minimum of *nnn*% of the scheduled system resources, which include CPU, storage, and paging capacity.
- `RELative nnnnn`: Specifies that this user is to receive a target minimum relative share of *nnnnn*. The amount of scheduled system resources that is available to relative share users is the total of resources that are available less the amount that is allocated to absolute share users.

cp set srm command

The **`cp set srm`** command sets some of the current z/VM system tuning parameters. These parameters define the size of the time slice, which is the access to the resources for different user classes as seen by the scheduler.

Each of these parameters has a default setting that is suitable for most environments. The following System Resource Manager (srm) parameters are included in this display:

- `iabias`: Interactive bias, used to favor short transactions.
- `ldubuf`: Controls z/VM's tolerance for guests that induce paging.
- `storbuf`: Can be used to encourage z/VM to overcommit main memory.
- `dispatching minor timeslice`: Controls the length of time that a single user holds onto a processor.
- `maxwss`: Sets the maximum working set that any normal user is allocated. It is specified as a percentage of the system's pageable real storage.
- `limithard`: Sets the enforcement of hard limiting of scheduled system resources. This setting affects only users with absolute maximum shares that are defined by using the **`SET SHARE`** command or the SHARE directory statement.

CP set quickdsp command

When the quick dispatch option is assigned to a virtual machine, that virtual machine is added to the dispatch list immediately, whenever it has work to do, without waiting in the eligible list. Instead of going into one of the usual queues, such as Q1, Q2, or Q3, they go into a special queue, Q0, from where they get dispatched as soon as possible. Figure 12-1 shows the output of the **`cp set quickdsp`** command.

```
set quickdsp mntlinux on
USER MNTLINUX:  QUICKDSP = ON
```

Figure 12-1 QUICKDSP command

Some useful **`cp set`** commands and their descriptions are listed in Table 12-2.

Table 12-2 CP SET commands

Set	Description
MDC	Use the SET MDC command from a Class B user to: ► Change minidisk cache settings for the entire system, for a real device, or for an active minidisk. ► Purge the cache of data from a real device or an active minidisk. ► Change a user's ability to insert data into the cache. This command can be useful when little minidisk activity exists and you want to release some storage.
QIOASSIST	Use SET QIOASSIST to control the queue-I/O assist (QDIO performance assist for V=V guests) for a virtual machine. This interpretive-execution assist applies to devices that use the Queued Direct I/O (QDIO) architecture, HiperSockets devices, and FCP devices.
RESERVED	Use SET RESERVED to establish the number of real storage frames that are available to a specific virtual machine. This might be useful if you are attempting to tune specific guests in a storage-constrained environment.

For more information about syntax and details about these commands, see the z/VM Help information or *z/VM 7.2 CP Command and Utilities Reference*.

12.1.3 Other basic and useful z/VM commands

Other useful basic commands are described in this section. All examples are shown from the MAINT virtual machine. The results differ for users with fewer privileges.

Getting help

To get help on the system, use the **HELP** command. Sometimes, it is difficult to find help for the exact command for which you are looking. The following **HELP** commands are useful:

```
===> help             // for basic help
===> help menus       // for menu of all z/VM help menus
===> help cp menu     // for a menu of all CP commands
===> help cpquery     // for a menu of all CP QUERY command
===> help cpset       // for a menu of all CP SET commands
```

Determining who is logged on

To see who is logged on to the system, use the **QUERY NAMES** command, for example:

```
===> q n
DIRMSAT2 - SSI
ZMAPVM62 - DSC , LINUX153 - DSC , LNXADMIN - DSC , LINUX157 - DSC
VSMEVSRV - DSC , VSMPROXY - DSC , VSMREQIU - DSC , VSMREQI6 - DSC
VSMREQIN - DSC , DTCSMAPI - DSC , PERSMAPI - DSC , VSMWORK3 - DSC
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
TCPIP    - DSC , DIRMAINT - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC
VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
MAINT    -L0004
VSM     - TCPIP
```

Determining storage or memory

To see how much main storage (memory) is installed and allocated to a system, use the **QUERY STORAGE** command, for example:

```
===> q stor
STORAGE = 16G CONFIGURED = 16G INC = 256M STANDBY = 0  RESERVED = 0
```

This example shows 16 GB of central memory (storage).

> **Note:** The results of this command are based on the CP Class your user ID has based on the role. If you are a System Admin, you might have classes A, B, and C and see the z/VM amount of storage (memory). However, if you have only class G, it shows the user ID virtual storage only.

Determining processors or CPUs

To see how many processors (central processors (CPs), Integrated Facilities for Linux (IFLs), and CPUs) are allocated at system level, use the **QUERY PROCESSORS** command, for example:

```
===> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
```

```
PROCESSOR 03 ALTERNATE CP
PROCESSOR 04 ALTERNATE CP
PROCESSOR 05 ALTERNATE CP
PROCESSOR 06 ALTERNATE CP
PROCESSOR 07 ALTERNATE CP
PROCESSOR 08 ALTERNATE CP
PROCESSOR 09 ALTERNATE CP
```

Determining the software level

To determine the control program (CP) level of your system, use the **QUERY CPLEVEL** command, for example:

```
===> q cplevel
z/VM Version 7 Release 2.0, service level 2001 (64-bit)
Generated at 07/29/20 16:50:40 EDT
IPL at 09/26/20 19:28:25 EDT
```

To determine the CMS level of your system, use the following **QUERY CMSLEVEL** command:

```
===> q cmslevel
q cmslevel
z/CMS Level 30, Service Level 0000
```

Determining the system cylinder allocation

The **QUERY ALLOC MAP** command shows you the system allocation of spool, paging, and directory space, for example:

```
===> q alloc map
                EXTENT     EXTENT                         % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
JV1030 1030          1         20     20      1      1   5% DRCT ACTIVE
JV1031 1031          1       3338 600840  87022  91029  14% SPOOL
JV1131 1131          -          -      0      0      0   0  SHARED
JP1260 1260          0      10016  1761K     27     56   1% PAGE
JP1261 1261          0      10016  1761K     75     75   1% PAGE
JV1032 1032          1       3338 600840     52     63   1% PAGE
```

Determining DASD, OSA, and virtual resources

The **QUERY DASD** and **QUERY DASD FREE** commands show you the DASD that is assigned to the system and free DASD that is available to be assigned. Similarly, the **QUERY OSA** and **QUERY OSA FREE** commands report on the Open Systems Adapter (OSA) resources.

Finally, the **QUERY VIRTUAL ALL** command can be useful when looking at the virtual resources of the user ID to which you are logged on. The following short forms of these commands are available without any of the associated output:

```
===> q da
===> q da free
===> q osa
===> q osa free
===> q v all
```

Note: You can always use HELP to check the syntax of the commands when logged onto CMS.

12.2 z/VM Performance Toolkit

To use the z/VM Performance Toolkit, you must order the product. Configure the product only if you ordered it. z/VM Performance Toolkit is part of the z/VM base installation, and it is installed as disabled. It is a priced feature of z/VM.

For more information, see the following publications:

- *Z/VM Performance Toolkit Guide*, SC24-6302
- z/VM Performance Toolkit Reference, SC24-6303
- *The Program Directory for Performance Toolkit for VM*, GI13-4361
- *Linux on IBM zSeries and S/390: Performance Toolkit for VM*, SG24-6059

For more information about how to set up and use the IBM Performance Toolkit, see the following sections:

- 12.2.1, "Configuring IBM Performance Toolkit for VM"
- 12.2.5, "Using the IBM Performance Toolkit for VM" on page 387

12.2.1 Configuring IBM Performance Toolkit for VM

IBM Performance Toolkit is installed with z/VM. The configuration is described in the *Program Directory for Performance Toolkit for z/VM*.

Complete the following steps to turn it on (configure the product only if you ordered it):

1. Query the priced products that are enabled by using the **QUERY PRODUCT** command:

```
===> q product
Product  State    Description
IBMVMSSI Enabled  IBM z/VM Single System Image Feature
7VMDIR20 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FACILITY
(Dir
Maint)
7VMPTK20 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
7VMRAC20 Disabled 00/00/00.00:00:00.$BASEDDR RACF Security Server
7VMRSC20 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
```

2. To enable IBM Performance Toolkit for VM, log on as **MAINT720** and enter the following command:

```
===> service perftk enable
VMFSRV2760I SERVICE processing started
...
VMFSRV1233I The following products have been serviced.
VMFSRV1233I PERFTKSFS
VMFSRV2264I Restoring prior system environment using saved access/minidisk
            information
VMFSET2760I VMFSETUP processing started for ENVRESTORE
            SERVICEEXEC20201012073613
VMFSET2760I VMFSETUP processing completed successfully (RC=0)
VMFSRV2760I SERVICE processing completed successfully (RC=0)
```

   You see a few panes of messages scroll by and finally the success messages appear. Performance Toolkit is enabled for the current z/VM session.

   This process modifies the `SYSTEM CONFIG` file by appending a line to the end. Verify that this line was added by using the following commands:

```
===> vmlink pmaint cf0
```

```
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
===> type system config z
...    // many screens cleared
PRODUCT PRODID 7VMPTK20 STATE ENABLED DESCRIPTION '10/12/20.07:36:14.MAINT720
PERFKIT Minidisk Install and Service'
```

The **QUERY PRODUCT** command shows the change:

```
===> q product
Product  State    Description
IBMVMSSI Enabled  IBM z/VM Single System Image Feature
7VMDIR20 Enabled  09/27/20.16:02:25.MAINT720 Install/service DirMaint using SFS
7VMPTK20 Enabled  10/12/20.07:36:14.MAINT720 PERFKIT SFS Install and Service
7VMRAC20 Enabled  10/11/20.17:27:42.MAINT720 RACF Feature of z/VM, FL720
7VMRSC20 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
```

The Performance Toolkit is now enabled. You can also verify it by running the **QUERY PRODUCT** command again.

12.2.2 Configuring web browser support

Use the following command to log on to the default TCPMAINT user ID:

```
LOGON TCPMAINT BY IBMVM1
```

You can change the default TCPMAINT user ID in the USER DIRECT file.

After the product is enabled, the TCP/IP profile must be modified to enable web access to the Performance Toolkit. The following example sets the port to 80, which is the default for a web browser:

1. Log on to **TCPMAINT.** Edit the TCPIP configuration file - the default name is PROFILE TCPIP and search for the string reserve ports (where z/VM TCP/IP ports are reserved):

   ```
   ===> x profile tcpip d
   ====> /port
   ```

2. Add the following line under the PORT entries:

   ```
   ...
   PORT
     20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
     21   TCP FTPSERVE            ; FTP Server
     23   TCP INTCLIEN            ; TELNET Server
   ; 25   TCP SMTP                ; SMTP Server
     80   TCP PERFSVM             ; Performance Toolkit
   ; 111  TCP PORTMAP             ; Portmap Server
   ; 111  UDP PORTMAP             ; Portmap Server
   ; 143  TCP IMAP                ; IMAP Server
   ...
   ```

   Save your changes by issuing the **file** command.

3. Make permanent configuration changes to TCP/IP by editing the PROFILE TCPIP file on the TCPIP configuration disk, TCPMAINT 198. You also can make TCP/IP changes dynamically.

   To make a configuration change to TCP/IP while it is running, the following options are available:

   – The **NETSTAT OBEY** command allows a simple command to be entered (such as a **START** or **STOP** of a device).

– For changes that involve multiple configuration commands or statements, the **OBEYFILE** command is required. This command reads a file that contains TCP/IP PROFILE statements and applies those statements to the running system. An obey file can add new configuration statements, and change statements that span multiple lines (such as AUTOLOG, PORT, and GATEWAY).

To make a complex TCP/IP change dynamically, create a file on TCPMAINT 198 that contains your required changes. You can use your PROFILE TCPIP file for guidance. If you are adding details to a statement, such as PORT or GATEWAY that spans multiple lines, you must include the entire existing statement not just the new details you want to add.

After you create the file, you can use the **OBEYFILE** command to tell TCP/IP to process the file by using the following command:

OBEYFILE fn ft

Where *fn* is the name of the file you created, and *ft* is the file type (if you used the file type TCPIP for your obey file, you do not need to specify it).

If you did not set up RACF on your system, you must provide the read password of the TCPMAINT 198 disk as an option to this command.

When you run the **OBEYFILE** command, you receive a response. If TCP/IP encountered any errors in processing your obey file, you receive a message saying that a file that includes more messages was sent to your reader. You must review this file to see whether your TCP/IP statements were processed correctly.

After you update TCP/IP dynamically, you must make the changes persistent. Apply the changes from your obey file to the permanent TCP/IP configuration by merging the updates in your obey file into the PROFILE TCPIP file.

> **Note:** Rather than create a separate file, you can update TCP/IP by using the `PROFILE TCPIP` file as the obey file. You receive several error messages for options that are defined or unchanged, but the updated settings are applied as requested. This method is recommended for more experienced operators only.

4. Run the **NETSTAT CLIENTS** command to verify your configuration. You want to see that the service that is named `PERFSVM` is a client. `PERFSVM` is shown after a few windows of output:

```
===> netstat clients
...
Client: PERFSVM                Authorization: {none}
Notes Handled: none
Last Touched:   0.00:00:52
Vmcf error count: 0
```

If you are configuring central monitoring in a single system image (SSI) cluster, it is enough to configure the web server only in one of the members. Central monitoring enables one member to monitor the other members of the SSI cluster.

12.2.3 Configure PERFSVM

Run the following command to log on to the default PERFSVM user ID:

```
LOGON PERFSVM BY IBMVM1
```

You can change the default PERFSVM user ID in the USER DIRECT file.

The `PERFSVM` virtual machine is the Performance Toolkit service machine. Complete the following steps to configure it:

1. Log on to **PERFSVM**. If you successfully enabled the product, enter a Performance Toolkit session and see the following text at the top of the pane:

   ```
   FCX001                Performance Toolkit for VM              Autoscroll 12
    FCXBAS500I Performance Toolkit for VM FL720  (64-bit)
    FCXBAS100I HMA storage 2048M.2048M is being used for temporary work area
   ...
    07:55:24 HCPMOF6229E Monitor event collection is already active.
    07:55:24 HCPMOG6229E Monitor sample collection is already active.
   ```

2. Press **F12** twice to get to a Conversational Monitor System (CMS) prompt.
3. Copy the default configuration files, which are on `PERFSVM`'s D disk, to your A disk:

   ```
   ===> copy * * d = = a
   ```

4. The main configuration file is `FCONX $PROFILE`. Edit that file and search for the string VMCF:

   ```
   ===> x fconx $profile
   ====> /VMCF
   ```

   This search takes you to line 149 where the next eight lines are comments that start with an asterisk (*). Complete the following changes:

   – Uncomment the second, fourth, sixth, and eighth lines by changing `*C` to **FC**.

   – Change maxconn from `10` to **100** on the fourth line so that you can raise the connections limit.

   – Change port `81` to **80** on the fourth line so that you can use a browser interface without needing to specify port 81 on the URL (with a `:81` suffix).

   The modified lines look similar to the lines that are shown in Example 12-3. Save your changes by using the **FILE** command.

   Example 12-3 Modifications to the FCONX $PROFILE.

   ```
   *    Following command activates VMCF data retrieval interface
   FC MONCOLL VMCF ON
   *    Define the maximum allowed number of Internet connections
   FC MONCOLL WEBSERV MAXCONN 100
   *    Define the timeout of inactive Internet connections in minutes
   FC MONCOLL WEBSERV TIMEOUT 30
   *    Following command activates Internet interface
   FC MONCOLL WEBSERV ON TCPIP TCPIP 80
   ...
   ====> file
   ```

   If you are configuring central monitoring in an SSI cluster, enable the four FC commands on only one member, which serves as a web server. On the other members, allow only the first FC statement (**FC MONCOLL VMCF ON**).

5. Create a remote data retrieval authorization file with your z/VM system identifier (replace ***RDBKZVMF*** with your system identifier) by using the following commands:

```
===> x fconrmt authoriz
====> a
RDBKZVMF PERFSVM  S&FSERV DATA
```

If you are configuring central monitoring in an SSI cluster, allow the member that serves as the web server to access the other members. The authorization file on a second member looks like the following example:

```
====> a 2

RDBKZVMG PERFSVM  DATA
RDBKZVMH PERFSVM  S&FSERV DATA
```

6. Create a system identification file that links your z/VM systems and PERFSVM to a special resource name (replace ***ZVM63A*** with your system identifier):

```
===> x fconrmt systems
====> a
RDBKZVMF PERFSVM  z/VM7.2 N FCXC1R01
```

If you are configuring central monitoring in an SSI cluster, also specify all other members. Ensure that each member uses a unique resource name. The first member might be **FCXC1R01**, the second member might be **FCXC1R02**, and so on:

```
RDBKZVMG PERFSVM  z/VM7.2 N FCXC1R01
ITSOZVMH PERFSVM  z/VM7.2 N FCXC1R02
RDBKZVMI PERFSVM  z/VM7.2 N FCXC1R03
RDBKZVMJ PERFSVM  z/VM7.2 N FCXC1R04
```

The system identification files on all members must be the same.

7. Set up a resource override for the default resource name (enter the resource name that you used in FCONRMT AUTHORIZ):

```
===> x ucomdir names
====> a 6
:nick.FCXRES00 :luname.*IDENT
            :tpn.FCXC1R01
            :security.SAME
:nick.FCXSYSTM :luname.*IDENT
            :tpn.FCXC1S01
            :security.SAME
```

If you are configuring central monitoring in an SSI cluster, specify resource override on each member. The second member uses **FCXC1R02** and **FCXC1S02**. The third member uses **FCXC1R03** and **FCXC1S03**. Also, the fourth member uses **FCXC1R04** and **FCXC1S04**.

8. Make CP start to collect performance data. Perform the following steps to start Performance Toolkit automatically after the IPL. These steps should be performed on AUTOLOG2 if you have RACF enabled and on AUTOLOG1 if you do not plan to use RACF. Complete the following steps:

   a. Log on to AUTOLOG2.

   b. Before you press **Enter** at the VM READ prompt, enter **acc (noprof** so that the **PROFILE EXEC** is not run:

```
LOGON AUTOLOG2
z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR, 0008 PRT,   NO PUN
```

```
LOGON AT 12:13:55 EDT THURSDAY 06/06/13
z/VM V7.2.0    2013-06-04 12:50
acc (noprof
Ready; T=0.01/0.01 12:14:01
```

c. Edit the profile exec in the following way:

```
===> x profile exec a
...
/*********************************************************************/
/* Customer processing can be added here                             */
/*********************************************************************/
"CP XAUTOLOG TCPIP"              /* Autolog TCPIP                    */
"CP SET MDC STOR OM 256M"        /* Limit minidisk cache in CSTOR    */
"CP SET SIGNAL SHUTDOWN 300"     /* Allow guests 5 min to shut down */
"CP XAUTOLOG LNXADMIN"           /* Start the Linux admin machine    */

"CP MONITOR SAMPLE ENABLE PROCESSOR" /* Setup CP MONITOR parameters  */
"CP MONITOR SAMPLE ENABLE STORAGE"
"CP MONITOR SAMPLE ENABLE USER ALL"
"CP MONITOR SAMPLE ENABLE I/O ALL"
"CP MONITOR SAMPLE ENABLE NETWORK"
"CP MONITOR SAMPLE ENABLE APPLDATA ALL"
"CP MONITOR SAMPLE ENABLE ISFC"
"CP MONITOR SAMPLE ENABLE SSI"

"CP MONITOR EVENT  ENABLE STORAGE"
"CP MONITOR EVENT  ENABLE I/O ALL"
"CP MONITOR EVENT  ENABLE NETWORK"
"CP MONITOR EVENT  ENABLE ISFC"
"CP MONITOR EVENT  ENABLE SSI"

"CP MONITOR SAMPLE INTERVAL 1 MIN"  /* Set sampling interval */

"CP XAUTOLOG PERFSVM"            /* Start Performance Toolkit        */
```

d. Save the file by using the following command:

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

> **Note:** If you do not plan to IPL before you try Performance Toolkit, run all **CP MONITOR** commands that you just added to the **PROFILE EXEC** file so that CP starts to collect performance data.

e. Log off from AUTOLOG2.

12.2.4 Starting the IBM Performance Toolkit for VM

To start the Performance Toolkit, complete the following steps:

1. Log on to the PERFSVM virtual machine.
2. Press **Enter** and the Performance Toolkit starts through the PROFILE EXEC:

```
FCX001       VM/ESA Full Screen Op. Console / Perf. Monitor      Autoscroll 12
  FCXBAS500I Performance Toolkit for VM FL710  (64-bit)
  13:01:47 FCXCMK442E CP MONITOR data collection already active - command
  ignored
```

```
    13:01:47 FCXOMC772I SEGOUT data collection is active.  Using segment: PERFOUT
    13:01:47 FCXAPP530I Connected to *IDENT for resource FCXC1R01
    13:01:47 FCXAPF530I Connected to *IDENT for resource FCXC1S01
    13:01:47 FCXAPP527I User PERFSVM connected on path 0005
    13:01:47 FCXAPC535I Connected to resource FCXC1R01 on path 0004, for S&F-Coll
    13:01:47 HCPMOF6229E Monitor event collection is already active.
    13:01:47 HCPMOG6229E Monitor sample collection is already active.

Command ===> disc
```

The Performance Toolkit is now configured and running.

12.2.5 Using the IBM Performance Toolkit for VM

The Performance Toolkit can be used with a web browser or 3270 interface.

Using a web browser interface

To use the web-enabled Performance Toolkit, complete the following steps:

1. Point a browser to your z/VM system, for example:

   `http://`***`9.12.7.11`***

2. You see a splash window and then, the Web Server Log-on window, as shown in Figure 12-2.

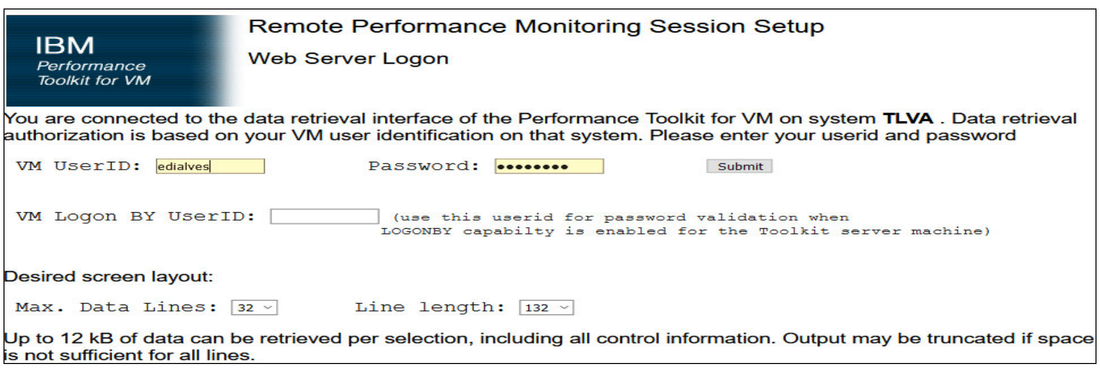


Figure 12-2 Performance Toolkit Web Server Log-on window

3. Enter any valid user ID and password and click **Submit** (in this example, `PERFSVM` was used).

   The Central Monitoring System Load Overview appears with your system identifiers (*Node-ID*) on the left side.

4. Click your system identifier and the Initial Performance Data Selection Menu window appears, as shown in Figure 12-3 on page 388.

   From this window, you can drill down into many different types of reports.

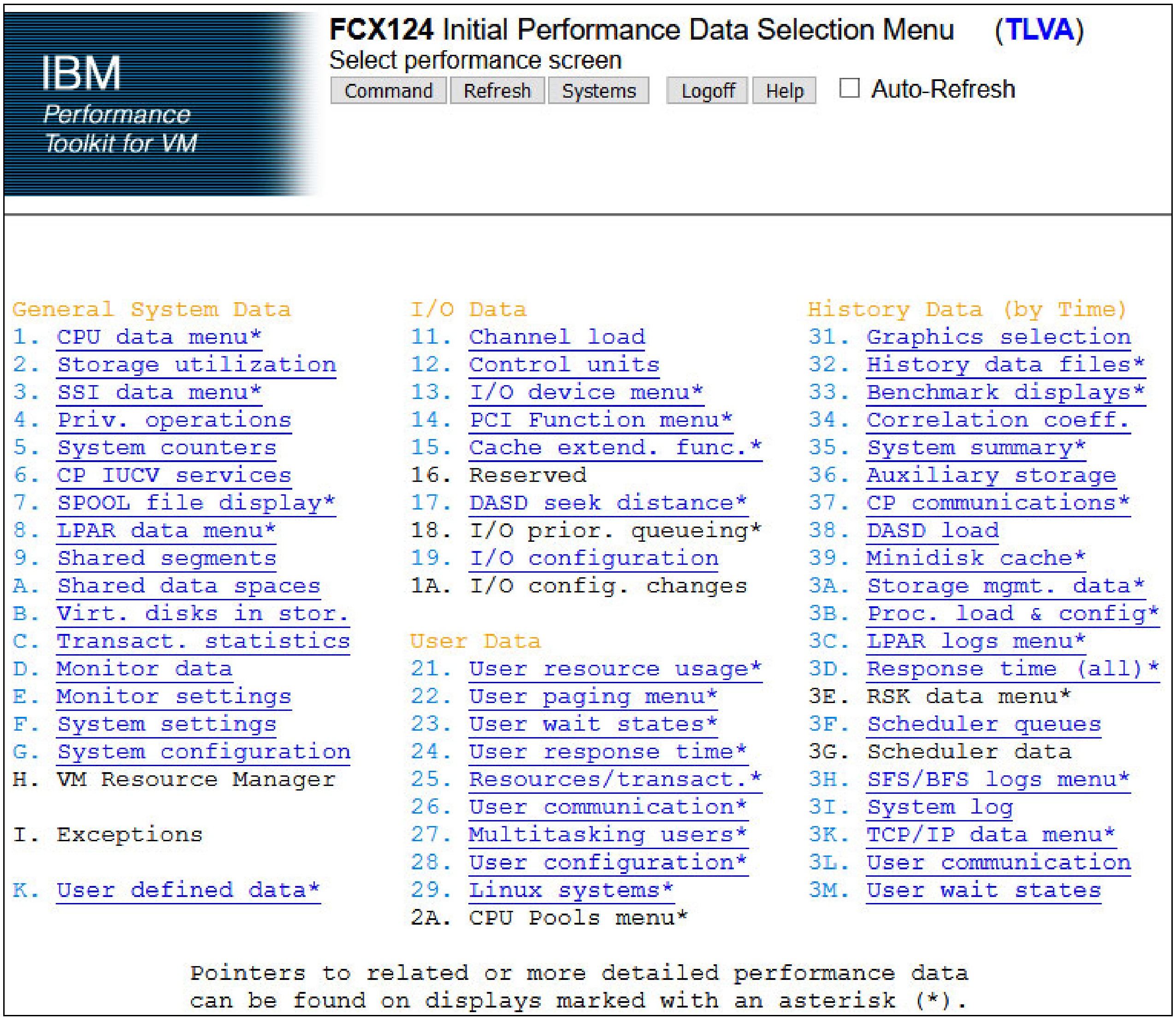


Figure 12-3 Using Browser interface to access Performance Toolkit menu

Using a 3270 interface

You can also use a 3270 interface. Complete the following steps:

1. Log on to PERFSVM.
2. If you are disconnected, pressing **Enter** returns you to the Performance Toolkit command line. If the virtual machine was logged off, the **PROFILE EXEC** runs and gets you to the command line. Enter the **MONITOR** command:

   ```
   Command ===> monitor
   ```

   The Performance Screen Selection pane then appears, as shown in Figure 12-4 on page 389.

```
FCX124          Performance Screen Selection  (FL720       )    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 menu*        33. Benchmark displays*
4. Priv. operations        14. Reserved                34. Correlation coeff.
5. System counters         15. Cache extend. func.*    35. System summary*
6. CP IUCV services        16. Reserved                36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*     37. CP communications*
8. LPAR data menu*         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. LPAR logs menu*
D. Monitor data            21. User resource usage*    3D. Response time (all)*
E. Monitor settings        22. User paging menu*       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
I. Exceptions              27. Multitasking users*     3K. TCP/IP data menu*
                           28. User configuration*     3L. User communication
K. User defined data*      29. Linux systems*          3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
```

Figure 12-4 Performance Screen Selection pane

Drilling down into report windows

You can now use the active report windows. To drill down into these windows, move the cursor to any of the titles that are active (active titles display the number or letter in white, and inactive titles are shown in green).

Several of the more useful report windows to drill down into are listed:

```
21. User resource usage
22. User paging load
23. User wait states
28. User configuration
29. Linux systems
33. Benchmark displays
```

12.3 Collecting and using raw CP monitor data

Although the Performance Toolkit formats and displays current performance data, it is often necessary to look at older data also. Typically, you compare the current system performance to the past performance so that data is available for troubleshooting, or to generate reports.

12.3.1 Collecting CP monitor data

CP monitor records are collected by the **MONWRITE** utility and written to a disk or tape. The resulting file contains all of the original unprocessed data. This data can be used later to generate reports or the Performance Toolkit can use this data in Monitor Data Scan Mode to look at historical data as though it was current:

1. Log on to the **MONWRITE** virtual machine.
2. Edit the **PROFILE EXEC**:

```
LOGON MONWRITE
z/VM Version 7 Release 2.0, Service Level 2001(64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:40:31 EDT FRIDAY 06/07/13
z/VM V7.2.0    2020-06-04 12:50

Ready; T=0.01/0.01 10:40:34
===> x profile exec a
input
/* ALL MONITOR COMMANDS ARE LOCATED IN AUTOLOG1'S PROFILE EXEC */
'MONWRITE MONDCSS *MONITOR DISK CLOSE 480'
===> file
```

3. Execute the REXX exec named **profile**:

```
===> profile
HCPMOW6272I Now recording in file D060713 T110146 A1
HCPMOW6265A MONITOR WRITER CONNECTED TO *MONITOR
```

4. Disconnect from **MONWRITE**:

```
===> #cp disc
```

The **CLOSE 480** statement tells **MONWRITE** to close the output file every 8 hours (480 minutes), starting from midnight. It means, regardless of when it starts recording, that it will close the file at 08:00, at 16:00, and at 24:00. The file name will clearly show the date and time when the recording started.

To collect **MONWRITE** data automatically, start the **MONWRITE** virtual machine when you IPL z/VM. Add a line to the **PROFILE EXEC** of the AUTOLOG1 191 disk (or AUTOLOG2 191 if an external security manager, such as RACF, is running):

```
===> x profile exec
...
"CP XAUTOLOG MONWRITE"              /* Start the MONWRITE VM       */
...
```

The **MONWRITE**'s A-disk is shipped as 300 cylinders, which is small. Depending on the monitor interval activity of the system and the number of samples/events, it can fill quickly. When the disk is full, **MONWRITE** will not be able to write anymore.

Important: Monitor the space on **MONWRITE**'s A-disk.

Another possibility is to use a utility that archives old files and cleans up the space automatically. MONCLEAN is an example of this type of utility.

You can download MONCLEAN from this web page.

Complete the following steps for the MONCLEAN installation:

1. Use FTP binary to transfer `MONCLEAN VMARC` to `MONWRITE`'s 191 disk.
2. Run `MONWRITE VMARC` through a **pipe** command:

   ```
   ===> pipe < monclean vmarc a | fblock 80 00 | > monclean vmarc A F 80
   ```

3. Unpack the `MONCLEAN VMARC` file with the **VMARC** command:

   ```
   ===> vmarc unpk monclean vmarc a
   MONCLEAN EXEC     A1. Bytes in=      4080, bytes out=      7678 (   188%).
   MONCLEAN README   A1. Bytes in=      1040, bytes out=      2240 (   215%).
   ```

4. Check the documentation in the `MONCLEAN README` file.
5. Modify PROFILE EXEC:

   ```
   ===> x profile exec
   /* ALL MONITOR COMMANDS ARE LOCATED IN AUTOLOG1'S PROFILE EXEC */
   'MONWRITE MONDCSS *MONITOR DISK CLOSE 60 EXEC MONCLEAN'
   ```

6. Start recording:

   ```
   ===> profile
   HCPMOW6272I Now recording in file D061213 T131724 A1
   HCPMOW6265A MONITOR WRITER CONNECTED TO *MONITOR
   ```

7. `MONWRITE` closes the output file every hour and runs `MONCLEAN EXEC`. If the `MONCLEAN EXEC` was not modified, it will remove the oldest file when the disk reaches 80% full.

   Example 12-4 shows `MONWRITE`'s 191 disk when **MONCLEAN** is running.

   Example 12-4 MONWRITE 191 disk

   ```
   MAINT    FILELIST A0   V 169  Trunc=169 Size=19 Line=1 Col=1 Alt=0
   Cmd    Filename Filetype Fm Format Lrecl     Records     Blocks   Date     Time
          D061313  T100016  Z1 F       4096       49275      49275  9/13/20 10:29:16
          D061313  T090016  Z1 F       4096       99407      99407  9/13/20 10:00:15
          D061313  T080015  Z1 F       4096       99392      99392  9/13/20  9:00:15
          D061313  T070015  Z1 F       4096       99348      99348  9/13/20  8:00:15
          D061313  T060015  Z1 F       4096       99348      99348  9/13/20  7:00:15
          D061313  T050016  Z1 F       4096       99348      99348  9/13/20  6:00:15
          D061313  T040016  Z1 F       4096       99348      99348  9/13/20  5:00:15
          D061313  T030015  Z1 F       4096       99348      99348  9/13/20  4:00:15
          D061313  T020016  Z1 F       4096       99348      99348  9/13/20  3:00:15
          D061313  T010015  Z1 F       4096       99348      99348  9/13/20  2:00:15
          D061313  T000015  Z1 F       4096       99348      99348  9/13/20  1:00:15
          D061213  T230015  Z1 F       4096       99348      99348  9/13/20  0:00:15
          PROFILE  EXEC     Z1 V         65           2          1  6/12/19 11:35:49
          MONCLEAN EXEC     Z1 V         75         194          2  6/12/19 11:32:13
          MONCLEAN README   Z1 F         80          28          1  6/12/19 11:32:13
          MONCLEAN VMARC    Z1 F         80          64          2  6/12/19 11:32:13
   ```

12.3.2 Using CP monitor data

With the Performance Toolkit subcommand **MONSCAN**, you can select a CP monitor file on disk or tape (that is created by the standard **MONWRITE** utility) as input for performance data analysis. When the specified file is located, a performance data scan mode is entered that looks almost identical to the normal real-time monitoring mode. You can use this mode to browse through the accumulated monitor data.

Because the `PERFSVM` virtual machine is used to show the current performance data, it is better to use a different virtual machine to perform **MONSCAN**. The following example uses the `MAINT` user ID.

Complete the following steps:

1. Link and access `PERFSVM`'s 201 minidisk:

```
===> vmlink perfsvm 201
DMSVML2060I PERFSVM 201 linked as 0120 file mode Z
```

2. Link and access `MONWRITE`'s 191 minidisk:

```
===> vmlink monwrite 191
DMSVML2060I MONWRITE 191 linked as 0121 file mode X
```

3. Check that the files are available from **MONWRITE**:

```
===> filel * * x
MAINT    FILELIST A0  V 169  Trunc=169 Size=4 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      D061020 T084824  X1 F       4096      53930      53930  6/10/20 9:20:43
      PROFILE EXEC X1 V        65          3          1  6/10/20 8:48:21
```

4. Run the **MONSCAN** subcommand:

```
===> perfkit monscan D061020 T084824 X
```

The regular Performance Screen Selection window opens (see Figure 12-5).

```
FCX124         Performance Screen Selection  (FL720       )    Monitor Scan

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 menu*        33. Benchmark displays*
4. Priv. operations       14. Reserved                34. Correlation coeff.
5. System counters        15. Cache extend. func.*    35. System summary*
6. CP IUCV services       16. Reserved                36. Auxiliary storage
7. SPOOL 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
I. Exceptions             27. Multitasking users*     3K. TCP/IP data menu*
                          28. User configuration*     3L. User communication
K. User defined data*     29. Linux systems*          3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
```

Figure 12-5   Performance Screen Selection window

5. Make a selection, for example, 1 - CPU load. The first window does not contain any data. Enter the **nexts** command (for next sample) and a window with real numbers opens. You can see the interval on the top of the window, as shown in Figure 12-6.

```
FCX100      Data for 2020/09/10  Interval 08:48:40 - 08:49:40    Monitor Scan

CPU Load                                          Status or
PROC TYPE %CPU  %CP %EMU %WT %SYS %SP %SIC %LOGLD  ded. User
P00  CP      0    0    0 100    0   0   99      0  Master
P01  CP      0    0    0 100    0   0   99      0  Alternate
P02  IFL     0    0    0 100    0   0  ...      0  Alternate
P03  IFL     0    0    0 100    0   0  ...      0  Alternate

Total SSCH/RSCH    254/s     Page rate       .0/s     Priv. instruct.    28/s
Virtual I/O rate    10/s     XSTORE paging   .0/s     Diagnose instr.    16/s
Total rel. SHARE    3050     Tot. abs SHARE    0%

Queue Statistics:     Q0      Q1      Q2      Q3     User Status:
VMDBKs in queue        1       0       1       0     # of logged on users      14
VMDBKs loading         0       0       0       0     # of dialed users          0
Eligible VMDBKs                0       0       0     # of active users          7
El. VMDBKs loading             0       0       0     # of in-queue users        2
Tot. WS (pages)     2911       0   41870       0     % in-Q users in PGWAIT     0
Reserved                                             % in-Q users in IOWAIT     0
85% elapsed time   96.00   16.00   128.0   768.0     % elig. (resource wait)    0

Transactions      Q-Disp    trivial   non-trv     User Extremes:
Average users        2.7         .8        .2     Max. CPU %   LNXADMIN      .1
Trans. per sec.       .2         .1        .0     Reserved
Av. time (sec)     18.40      12.39     16.39     Max. IO/sec  MONWRITE     9.4
UP trans. time                 .000      .000     Max. PGS/s   ........   .....
MP trans. time                12.39     16.39     Max. RESPG   LNXADMIN   41923
System ITR (trans. per sec. tot. CPU)    31.3     Max. MDCIO   MONWRITE      .1
Emul. ITR (trans. per sec. emul. CPU)   269.2     Max. XSTORE  ........   .....
```

Figure 12-6 CPU load performance

12.4 Monitoring Linux performance for troubleshooting

Previous sections described how the Performance Toolkit can show the resource consumption of the Linux guest as measured and dispatched by the z/VM hypervisor. z/VM is not aware of the nature of the guest and it cannot understand what is happening inside the guest. For that reason, it is important that you can measure performance data from within the Linux guest itself.

To monitor Linux performance data at this level, a data gatherer process must be running within each Linux guest that you want to monitor. Different ways of gathering this data are available. Many commercial and non-commercial solutions exist for long-term monitoring, also.

This book cannot cover all of the requirements for long-term monitoring (low CPU consumption, data storage, and so on). This chapter shows how to monitor Linux performance for short periods, especially when you are troubleshooting performance problems.

12.4.1 Monitoring Linux performance from z/VM

This section describes how to gather Linux performance data in Linux and provide this data to z/VM for a consolidated overview.

To monitor Linux performance data directly from the kernel, the following statements must be true:

- The `APPLMON` option must be set in the user directory.
- Applmon data monitoring must be built into the kernel.

The first requirement typically is true because the `OPTION APPLMON` was set for the Linux virtual machines in earlier sections. For the second requirement, this feature is built into both Red Hat Enterprise Linux (RHEL) Server and SUSE Linux Enterprise Server (SLES).

Complete the following steps to use this built-in monitoring function:

1. Start a Secure Shell (SSH) session to a Linux system. In this example, `LINUX3` is used.
2. Three modules are built into the kernel but they are not loaded, by default. They are named `appldata_mem`, `appldata_os`, and `appldata_net_sum`. You can verify that they are not loaded with the **`lsmod`** and **`grep`** commands:

```
# lsmod | grep appldata
```

3. No output results from the commands, so no modules with the string `appldata` are loaded. Load those modules by using the **`modprobe`** command and verify that they were loaded:

```
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

4. If you repeat the **`lsmod`** command, you will see the following output:

```
# lsmod | grep appldata
appldata_net_sum       1966  0
appldata_os            2989  0
appldata_mem           2008  0
```

The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory, five files exist:

`timer`	Controls whether any data gathering is in effect
`interval`	Sets the interval, in milliseconds, during which samples are taken
`mem`	Controls the memory data gathering module
`os`	Controls the CPU data gathering module
`net_sum`	Controls the net data gathering module

5. To turn on the built-in kernel monitoring, use the **`echo`** command to send a nonzero value into four of the five monitoring variables in the `/proc/` virtual file system:

```
# echo 1 > /proc/sys/appldata/timer
# echo 1 > /proc/sys/appldata/mem
# echo 1 > /proc/sys/appldata/os
# echo 1 > /proc/sys/appldata/net_sum
```

Built-in kernel monitoring is now turned on. You might want to leave only the monitoring on for specific periods of time. While Linux monitoring data is captured, the Performance Toolkit's minidisk space can fill up quickly.

Viewing performance data from the Linux kernel in the Performance Toolkit

After the system has time to collect data, you can use the Performance Toolkit to view Linux performance data. To view that data, drill down into menu 29, Linux systems. Use either the browser interface or the 3270 interface, as shown in Figure 12-7.

```
FCX242      CPU 3906 SER 23BD5          Linux Displays        Perf. Monitor



   Linux screens selection
 S Display     Description
 . LINUX       RMF PM system selection menu
 . LXCPU       Summary CPU activity display
 S LXMEM       Summary memory util. & activity display
 . LXNETWRK    Summary network activity display
```

Figure 12-7 Linux Displays

Then, type `S` over the period on the left side of the submenu window in the row that corresponds to the report that you want to see. You will see a new report window with the Linux guest systems memory overview, as shown in Figure 12-8.

```
FCX244      CPU 3906 SER 23BD5       Initial  14:22:57        Perf. Monitor
______           .      .       .      .      .       .      .       .      .
          <------------ Memory Allocation (MB) -------------> <------- Swapping
Linux     <--- Main ---> <--- High --->        Buffers  Cache <-Space (MB)-> <-
Userid    M_Total %MUsed H_Total %HUsed Shared /CaFree   Used S_Total %SUsed
>System<    491.6   25.8      .0     .0     .0     8.6   46.3   761.6     .0
LINUX3      491.6   25.8      .0     .0     .0     8.6   46.3   761.6     .0
```

Figure 12-8 Linux guest systems memory overview

You can also use a web interface to view the same data.

13

Disk storage administration

This chapter describes working with disks. Extended count key data (ECKD) DASD and Fibre Channel Protocol (FCP)/Small Computer System Interface (SCSI) tasks are described.

This chapter includes the following topics:

13.1 Adding disk space to Linux virtual machines

This section describes how to add disk space to a Linux virtual machine. This disk space might come from different types of disks. The types of disk are described in 2.4, "Disk planning" on page 43.

> **Important**: If you add minidisks to the user directory for a specific virtual machine, they can be attached to a running Linux system without "bouncing" it.
>
> For example, if you added a minidisk at virtual address `0704`, you can use the following commands to link to the disk and then, enable it:
>
> ```
> $ sudo vmcp link '* 0704 0704 mr'
> $ sudo chccwdev -e 0704
> ```

13.1.1 Making new minidisks or count key data DASD available in Linux

After making the required changes in the z/VM user directory to add a new minidisk or full-volume DASD to a Linux virtual server, the steps in this section describe the required tasks in Linux to bring those volumes online and incorporate them into the Linux system.

In our example environment, we added new volumes by using the addresses `0.0.0702`, `0.0.0703`, and `0.0.0704`.

Complete the following steps to make the new disks available for use:

1. Make the disks visible by using the **`cio_ignore -r`** command. The device numbers are removed from the I/O device blacklist, which makes it visible to Linux for enumeration and use:

   ```
   # cio_ignore -r 0702
   # cio_ignore -r 0703
   # cio_ignore -r 0704
   ```

2. Depending on your operating system, complete the following steps:

   – If you use Red Hat Enterprise Linux Server 8 (RHEL):

     i. Enable the disks by using the **`chccwdev -e`** command:

     ```
     # chccwdev -e 0702 103 104
     Setting device 0.0.0702 online
     Done
     Setting device 0.0.0703 online
     Done
     Setting device 0.0.0704 online
     Done
     ```

     ii. Make a backup of `/etc/dasd.conf`, and then, add minidisks `0702`, `103`, and `104` to it:

     ```
     # cd /etc
     # cp dasd.conf dasd.conf.orig
     # vi dasd.conf
     0.0.0901
     0.0.0900
     0.0.0701
     0.0.0700
     0.0.00702
     0.0.0703
     ```

```
0.0.0704
```

– If you use SUSE Linux Enterprise Server (SLES) 15, use the **dasd_configure** command to enable minidisks 0702, 103, and 104:

```
# dasd_configure 0.0.0702 1
Configuring device 0.0.00702
Setting device online
# dasd_configure 0.0.0703 1
Configuring device 0.0.0703
Setting device online
# dasd_configure 0.0.0704 1
Configuring device 0.0.0704
Setting device online
```

3. View the available disks again by using the **lsdasd** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
================================================================================
0.0.0901   active      dasda     94:0    FBA   512    512MB     1048576
0.0.0900   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0700   active      dasdc     94:8    ECKD  4096   3521MB    901440
0.0.0701   active      dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0702   n/f         dasde     94:16   ECKD
0.0.0703   n/f         dasdf     94:20   ECKD
0.0.0704   n/f         dasdg     94:24   ECKD
```

4. Format the disks in parallel with the **dasdfmt** command by using a **for** loop and putting them in the background:

```
# for i in e f g
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 1923
[2] 1924
[3] 1925
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok

[1]   Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[2]-  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[3]+  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
```

5. Create one partition from each of the disks by using a bash **for** loop and the **fdasd -a** command:

```
# for i in e f g
> do
>   fdasd -a /dev/dasd$i
> done
reading volume label ..: VOL1
reading vtoc ..........: ok
auto-creating one partition for the whole disk...
...
```

The three new minidisks are now low-level formatted, partitioned, and configured to be active at start time.

If you are creating a logical volume, see 13.2.1, “Creating a logical volume and file system” on page 404. If you are extending an existing logical volume, see 13.3, “Extending a logical volume” on page 408.

13.1.2 Making new emulated DASD available in Linux

For new emulated DASD (EDEV), for example, at address `0.0.0750`, that is dedicated to your system, the procedure to integrate them into the system is similar to the procedure for CKD DASD. The main difference is the tool that is used to format and partition these disks.

Complete the following steps:

1. Make the disk visible by using the **cio_ignore** command:

   ```
   # cio_ignore -r 0750
   ```

2. Depending on your operating system, complete the following steps:

   – If you use Red Hat Enterprise Linux 8, follow these steps:

     i. Enable the disks by using the **chccwdev -e** command:

     ```
     # chccwdev -e 0750
     Setting device 0.0.0750 online
     Done
     ```

     ii. Make a backup of `/etc/dasd.conf`, and then, add the DASD `0750` to it:

     ```
     # cd /etc
     # cp dasd.conf dasd.conf.orig
     # echo 0.0.0750 >> dasd.conf
     ```

   – If you use SUSE Enterprise Linux 15, use the **dasd_configure** command to enable the DASD `0750`:

     ```
     # dasd_configure 0.0.0750 1
     Configuring device 0.0.0750
     Setting device online
     ```

3. View the available disks again by using the **lsdasd** command:

   ```
   # lsdasd
   Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
   ==============================================================================
   0.0.0901   active      dasda     94:0    FBA   512    512MB     1048576
   0.0.0900   active      dasdb     94:4    FBA   512    256MB     524288
   0.0.0700   active      dasdc     94:8    ECKD  4096   3521MB    901440
   0.0.0701   active      dasdd     94:12   ECKD  4096   3521MB    901440
   0.0.0750   active      dasde     94:16   FBA   512    10240MB   20971520
   ```

4. Create a partition on the disk by using the **parted** command:

   ```
   # parted -s /dev/dasde mklabel msdos mkpart primary 0% 100%
   ```

The new DASD is now partitioned, and it is configured to be active at start time.

If you are creating a logical volume, see 13.2.1, “Creating a logical volume and file system” on page 404. If you are extending an existing logical volume, skip ahead to 13.3, “Extending a logical volume” on page 408.

13.1.3 Making new zFCP LUN available in Linux

To use Fibre Channel Protocol (FCP) in a single system image (SSI) environment, you must understand that within Linux, you need to handle more adapters than the adapters that are visible in only one SSI node. Fortunately, both SLES 15 and RHEL 8 changed the behavior of FCP to automatic logical unit number (LUN) detection. Therefore, it is sufficient to merely configure the host adapters and use the multipathed device only for disk configurations.

This section assumes that no previous zFCP was available. The planning according to this manual creates two FCP adapters at the addresses `0.0.fc00` and `0.0.fd00`. The necessary setup for z/VM is described in detail in 11.2.2, “Direct-attached Fibre Channel” on page 348. Follow these steps:

1. Start a Secure Shell (SSH) session to the target system.
2. Check that two devices are available with the **CP QUERY FCP** command:

```
#  vmcp q v fcp
FCP  FC00 ON FCP   B801 CHPID 70 SUBCHANNEL = 0001
     FC00                        TOKEN      = 00000007F62EA280
     FC00 DEVTYPE FCP         VIRTUAL CHPID FF FCP REAL CHPID 70
     FC00 QDIO ACTIVE         QIOASSIST ACTIVE        QEBSM
     FC00
     FC00 INP + 01 IOCNT = 00001346  ADP = 128 PROG = 000 UNAVAIL = 000
     FC00          BYTES = 0000000000000000
     FC00 OUT + 01 IOCNT = 00001464  ADP = 000 PROG = 128 UNAVAIL = 000
     FC00          BYTES = 00000000005711FE
     FC00 DATA ROUTER ACTIVE
     WWPN C05076DD90000404
FCP  FD00 ON FCP   B901 CHPID 71 SUBCHANNEL = 0002
     FD00                        TOKEN      = 00000007F62EA380
     FD00 DEVTYPE FCP         VIRTUAL CHPID 71 FCP REAL CHPID 71
     FD00 QDIO ACTIVE         QIOASSIST ACTIVE        QEBSM
     FD00
     FD00 INP + 01 IOCNT = 00001338  ADP = 128 PROG = 000 UNAVAIL = 000
     FD00          BYTES = 0000000000000000
     FD00 OUT + 01 IOCNT = 00001428  ADP = 000 PROG = 128 UNAVAIL = 000
     FD00          BYTES = 000000000052EF86
     FD00 DATA ROUTER ACTIVE
     WWPN C05076DD90000A64
```

3. Make the disk visible with the **cio_ignore** command:

```
# cio_ignore -r fc00
# cio_ignore -r fd00
```

If you use Red Hat Enterprise Linux 8, follow these steps:

1. Enable the FCP adapters by using the **chccwdev** command:

```
# chccwdev -e fc00
Setting device 0.0.fc00 online
Done
# chccwdev -e fd00
Setting device 0.0.fd00 online
Done
```

2. Verify that the auto LUN scan feature detected all of the paths to the LUNs:

```
# lsluns
Scanning for LUNs on adapter 0.0.fc00
        at port 0x500507630500c74c:
                0x4010401700000000
        at port 0x50050763050bc74c:
                0x4010401700000000
Scanning for LUNs on adapter 0.0.fd00
        at port 0x500507630510c74c:
                0x4010401700000000
        at port 0x50050763051bc74c:
                0x4010401700000000
```

3. If multipath is not yet configured, complete the following steps:

   a. Install the `device-mapper-multipath`:

   ```
   # yum -y install device-mapper-multipath
   Installed:
      device-mapper-multipath-0.4.9-77.el7.s390x
      ...
   ```

   b. Copy the multipath reference configuration file to the `/etc/multipath.conf` file:

   ```
   # cp  /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf
   /etc/multipath.conf
   ```

   c. Check the status of the multipathd daemon. If it is not started, start the service and then make it permanent:

   ```
   # systemctl status  multipathd
   multipathd.service - Device-Mapper Multipath Device Controller
      Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled)
      Active: active (running) since Wed 2015-04-29 08:42:02 EDT; 25s ago
     Process: 2962 ExecStart=/sbin/multipathd (code=exited, status=0/SUCCESS)
     Process: 2958 ExecStartPre=/sbin/multipath -A (code=exited,
   status=0/SUCCESS)
     Process: 2953 ExecStartPre=/sbin/modprobe dm-multipath (code=exited,
   status=0/SUCCESS)
    Main PID: 2965 (multipathd)
      CGroup: /system.slice/multipathd.service
              /sbin/multipathd
   # systemctl start multipathd
   # systemctl enable multipathd
   ```

   d. Verify whether multipath set the correct paths to the LUN:

   ```
   # multipath -ll
   mpatha (36005076305ffc74c0000000000001017) dm-2 IBM     ,2107900
   size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
   `-+- policy='service-time 0' prio=1 status=active
     |- 0:0:0:1075265552 sda 8:0  active ready running
     |- 0:0:1:1075265552 sdb 8:16 active ready running
     |- 1:0:0:1075265552 sdc 8:32 active ready running
     `- 1:0:1:1075265552 sdd 8:48 active ready running
   ```

4. Make the FCP configuration persistent:

```
# lszfcp -D | awk '{ print $1 }' | sed -e 's/\// /g' >> /etc/zfcp.conf
```

5. Create a partition on the multipath device by using the **parted** command:

```
# parted -s /dev/mapper/mpatha mklabel msdos mkpart primary 0% 100%
```

If you use SUSE Enterprise Linux 15, follow these steps:

1. Enable the FCP adapters' **zfcp_host_configure** command:

```
# zfcp_host_configure 0.0.fc00 1
# zfcp_host_configure 0.0.fd00 1
```

2. Verify that the auto LUN scan feature detected all of the paths to the LUNs:

```
# lsluns
Scanning for LUNs on adapter 0.0.fc00
        at port 0x500507630500c74c:
                0x4010401700000000
        at port 0x50050763050bc74c:
                0x4010401700000000
Scanning for LUNs on adapter 0.0.fd00
        at port 0x500507630510c74c:
                0x4010401700000000
        at port 0x50050763051bc74c:
                0x4010401700000000
```

3. Set up a multipath configuration if it is not configured:

   a. Ensure that the `multipath-tools` RPM is installed with the following **zypper** command:

   ```
   # zypper in multipath-tools
   ```

   b. Run the multipath daemon:

   ```
   # systemctl enable multipathd
   # systemctl start multipathd
   ```

   c. Create a partition on the disk by using the **parted** command:

   ```
   # parted -s /dev/mapper/mpatha mklabel msdos mkpart primary 0% 100%
   ```

   d. Use YaST to set up the partitioning for the multipath device. In this case, the FCP disk will become a LUN in a Logical Volume Manager (LVM) group for the `/srv/` directory:

   ```
   Run yast -> System -> Partitioner.
   Click Yes if you are asked if you really want to use this tool.
   Select System View -> Hard Disks and press [+].
   There is a new device available that represents the multipathed FCP disks.
   Add a partition that covers the full disk. Use Raw Volume and Do not format
   partition and Do not mount partition.
   Select System View -> Volume Management.
   Click Add -> Volume Group.
   Use vg_srv as Volume Group Name.
   Add the device to the volume group.
   Click Finish.
   Select System View -> Volume Management.
   Click Add -> Logical Volume.
   Set the name of the Logical Volume to srv, click Next.
   Use Maximum Size and click Next.
   Select Format partition and use file system XFS.
   Select Mount partition and set the Mount Point to /srv.
   Click finish and Next.
   Click Finish and leave YaST with Quit.
   ```

e. Check whether all paths are online:

```
# multipath -ll
36005076305ffc74c0000000000001119 dm-4 IBM,2107900
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 0:0:0:1075396625 sda 8:0  active ready running
  |- 0:0:1:1075396625 sdb 8:16 active ready running
  |- 1:0:0:1075396625 sdc 8:32 active ready running
  `- 1:0:1:1075396625 sdd 8:48 active ready running
```

4. To activate a new LUN to an existing volume group, run the following command:

```
# rescan_scsi_bus -a
```

13.2 Adding a logical volume

Sometimes, you require more disk space than a single DASD volume provides. For example, if you want a shared /home/ directory, it must be a sufficient size for many users to write data to. You can use the LVM to combine multiple DASD volumes into one logical volume. This example does not create a large logical volume, but it shows all of the necessary steps.

The following sections describe a logical volume with additional DASD on a Linux guest. Use the following overall steps in adding a logical volume.

13.2.1 Creating a logical volume and file system

The following overall steps are involved in creating a logical volume:

1. Create physical volumes from the two partitions.
2. Create a single volume group.
3. Create a single logical volume.
4. Make a file system from the logical volume.

Figure 13-1 on page 405 shows a block diagram of the LVM.

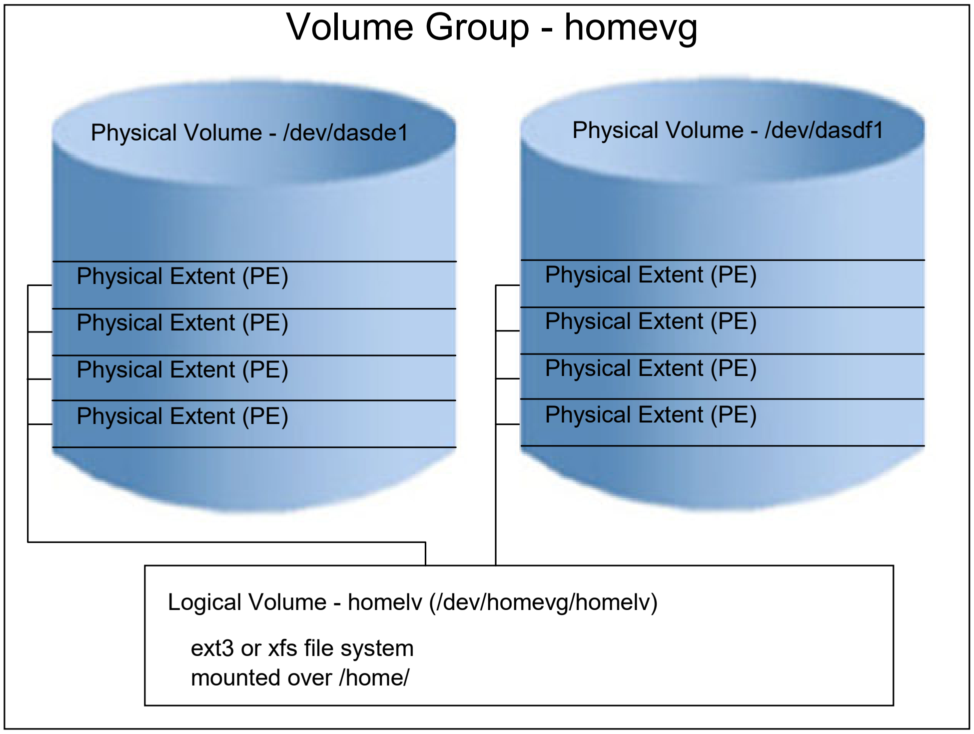


Figure 13-1 LVM block diagram

Creating physical volumes from two minidisks

To create physical volumes from new minidisks at virtual device addresses `0702` and `103`, completing the following steps:

1. Check the devices on your system with the **`lsdasd`** command.
2. The **`pvcreate`** command initializes partitions for use by LVM. Initialize the two new DASD partitions:

```
# pvcreate /dev/dasde1 /dev/dasdf1
  Physical volume "/dev/dasde1" successfully created
  Physical volume "/dev/dasdf1" successfully created
```

3. Verify that the physical volumes were created with the **`pvdisplay`** command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
  "/dev/dasde1" is a new physical volume of "3.44 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/dasde1
  VG Name
  PV Size               3.44 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               sOugfl-hlV3-fYnf-1adW-4mOI-4HTJ-HdAOTU

  "/dev/dasdf1" is a new physical volume of "3.44 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/dasdf1
```

```
     VG Name
     PV Size               3.44 GiB
     Allocatable           NO
     PE Size               0
     Total PE              0
     Free PE               0
     Allocated PE          0
     PV UUID               vO2PJY-gy4x-M9Hj-kt51-TO4J-B4n5-Ntvkje
```

Creating a single volume group

The **vgcreate** command is used to create a volume group that is named homevg from the two partitions. Use the **vgdisplay homevg** command to verify that the volume group was created:

```
# vgcreate homevg /dev/dasde1 /dev/dasdf1
Volume group "homevg" successfully created
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               6.88 GiB
  PE Size               4.00 MiB
  Total PE              1760
  Alloc PE / Size       0 / 0
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               acSF65-56Ie-kVoY-Af6I-Hma4-VVuN-ggJEs5
```

This example uses 1,760 free physical extents (PEs).

Creating a single logical volume

In this section, you create a single logical volume by using the **lvcreate** command:

1. The **lvcreate** command is used to create a logical volume. The **-i** (a lowercase I character) flag specifies the number of stripes, which is two in this example, because two volumes are in the volume group. The **-l** (a lowercase L) flag specifies the number of logical extents, which is 1,760 in this example. The **-n homelv** specifies the name of the new logical volume. The last argument, which is **homevg**, specifies the name of the volume group from which the logical volume will be created:

   ```
   # lvcreate -i 2 -l 1760 -n homelv homevg
     LUsing default stripesize 64.00 KiB
     Logical volume "homelv" created
   ```

2. Use the **lvdisplay** command to verify. The parameter is the full path of the logical volume, not the logical volume name:

```
# lvdisplay /dev/homevg/homelv
  --- Logical volume ---
  LV Path                /dev/homevg/homelv
  LV Name                homelv
  VG Name                homevg
  LV UUID                qNcyDp-Eeqs-gfBl-XU5Z-Jt3K-QfvV-pf3Kos
  LV Write Access        read/write
  LV Creation host, time virtcook3.itso.ibm.com, 2013-06-17 15:32:39 -0400
  LV Status              available
  # open                 0
  LV Size                6.88 GiB
  Current LE             1760
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     512
  Block device           253:4
```

Making a file system from the logical volume

Create a file system from the new logical volume.

If you are on RHEL 6.4, `ext4` is the recommended file system. Create an `ext4` file system on the new logical volume by using the **mkfs.ext4** command:

```
# mkfs.ext4 /dev/homevg/homelv
...
This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

If you are on SLES, `xfs` is the recommended file system for data. Use the following command to make the file system:

```
# mkfs.xfs /dev/homevg/homelv
...
```

The file system that was created from the logical volume is now ready to be mounted.

13.2.2 Updating the file system table

You can mount the file system manually. However, if you add the mount to the file system table file, `/etc/fstab`, you can effectively test the change by using the **mount** command with only one argument. Perform the following steps:

1. Make a backup copy of the file and add the following line to it:

```
# cd /etc
# cp fstab fstab.works
```

2. Add one line to the `fstab` file:

```
# vi fstab
... // For RHEL 6.4:
/dev/homevg/homelv      /home                   ext4    defaults        0 0
... // For SLES:
/dev/homevg/homelv      /home                   xfs     defaults        0 0
...
```

3. Before you mount over `/home/`, you might want to check that it is empty. If a non-root user exists and a new file system is mounted over it, the contents of the directory will be *hidden*. In this example, no data is in the file system:

```
# ls -a /home
.  ..
```

4. Mount the `/home/` file system with one argument. By using only one argument, you are testing the change to the file system table file, `/etc/fstab`. Use the **df -h** command to verify that it is mounted:

```
# mount /home
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdc1          1008M  184M  774M  20% /
tmpfs                 246M     0  246M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.3G  617M  68% /usr
/dev/mapper/system_vg-var_lv
                      504M   92M  388M  20% /var
/dev/mapper/homevg-homelv
                      6.8G  144M  6.3G   3% /home
```

5. Test a reboot to verify that the new logical volume is successfully mounted over `/home/`:

```
# reboot
Broadcast message from root@virtcook3.itso.ibm.com
        (/dev/pts/0) at 15:51 ...

The system is going down for reboot NOW!
```

When the system comes back, you will see the new logical volume that is mounted over `/home/`.

13.3 Extending a logical volume

This section describes the process of adding a minidisk to an existing LVM. This process is useful when your logical volume runs out of space. In this example, the `/var/` file system is filling up on `LINUX3`:

```
# df -h /var/
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/system_vg-var_lv
                      504M  392M   88M  82% /var
```

A 3390-9 was added as minidisk `106` in section 13.1, “Adding disk space to Linux virtual machines” on page 398.

> **Important:** You can attach minidisks to a running Linux system without rebooting the Linux system. For example, if you added a minidisk at virtual address `106`, from a root SSH session, use the **`vmcp link * 106 106 mr`** command to link to the minidisk. Then, use the **`chccwdev -e 106`** command to enable it.

To extend the logical volume by using this disk, perform the following steps:

1. Use the **vgdisplay** command to see the free space in the volume group `system_vg`:

```
# vgdisplay system_vg
  --- Volume group ---
  VG Name               system_vg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  6
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                5
  Open LV               4
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               5.88 GiB
  PE Size               4.00 MiB
  Total PE              1504
  Alloc PE / Size       1504 / 5.88 GiB
  Free  PE / Size       0 / 0
  VG UUID               4i89gF-b0xm-dkHo-blWP-3Kca-0xCI-V6TAXk
```

   This output shows no free extents in the volume group.

2. Use the **lsdasd** command to show the enabled disks:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0700   active      dasda     94:0    ECKD  4096   3521MB    901440
0.0.0901   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0900   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0701   active      dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0702   active      dasde     94:16   ECKD  4096   3521MB    901440
0.0.0703   active      dasdf     94:20   ECKD  4096   3521MB    901440
0.0.0704   active      dasdg     94:24   ECKD  4096   7042MB    1802880
```

   This output shows that minidisk `104` is at `/dev/dasdg`.

3. Make minidisk `104` a physical volume with the **pvcreate** command:

```
# pvcreate /dev/dasdg1
  Physical volume "/dev/dasdg1" successfully created
```

4. Use the **vgextend** command to add the minidisk to the volume group:

```
# vgextend system_vg /dev/dasdg1
  Volume group "system_vg" successfully extended
```

5. Use the **vgdisplay** command again to show the free extents in the volume group:

```
# vgdisplay system_vg
  --- Volume group ---
  VG Name               system_vg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  7
```

```
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                5
  Open LV               4
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               12.75 GiB
  PE Size               4.00 MiB
  Total PE              3264
  Alloc PE / Size       1504 / 5.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               4i89gF-b0xm-dkHo-blWP-3Kca-OxCI-V6TAXk
```

This output shows that 1,760 free extents are in the volume group now.

6. Use the **mount** command to determine the name of the logical volume that is mounted over /var/:

```
# mount | grep "\/var "
/dev/mapper/system_vg-var_lv on /var type ext4 (rw)
```

   In this example, the name is /dev/mapper/system_vg-var_lv/.

7. Use the **lvextend** command to extend the volume group with all of the new extents:

```
# lvextend -l +1760 /dev/mapper/system_vg-var_lv
  Extending logical volume var_lv to 7.38 GiB
  Logical volume var_lv successfully resized
```

8. Use the **resize2fs** command to increase the size of the ext4 file system while it is still mounted:

```
# resize2fs /dev/mapper/system_vg-var_lv
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/system_vg-var_lv is mounted on /var; on-line resizing
required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/mapper/system_vg-var_lv to 1933312 (4k)
blocks.
The filesystem on /dev/mapper/system_vg-var_lv is now 1933312 blocks long.
```

9. Use the **xfs_growfs** command to increase the size of the XFS file system while it is still mounted:

```
# xfs_growfs /dev/mapper/system_vb-var_lv
```

10. Use the **df** command to show the file system size before and after you extend it, as shown in the following example:

```
# df -h /var
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/system_vg-var_lv
                      7.3G  393M  6.6G   6% /var
```

   This output shows that the /var/ file system now has 6.6 GB of free space.

13.4 Moving a physical volume

In addition to file systems that grow larger, you might need to move data off one or more volumes on to another or a target set of volumes. If your data is in LVM, the **pvmove** and **vgreduce** commands were designed for this process, and they can be used with the file system online.

In this example, two physical volumes, /dev/dasde1 and /dev/dasdf1, exist. Data is populated on the first volume, and later moved to the second volume. This movement is performed while the file system is online.

To complete this test, perform the following steps:

1. Create a volume group from the first logical volume. In this example, it is named homelv:

```
# vgcreate homevg /dev/dasde1
  Volume group "homevg" successfully created
```

2. Observe the number of physical extents:

```
# vgdisplay homevg | grep "Total PE"
  Total PE              1760
```

3. Create a logical volume from the volume group. In this example, it is named homelv and all physical extents are used:

```
# lvcreate -l 1760 -n homelv homevg
  Logical volume "homelv" created
```

4. Create a file system from the logical volume. In this example, it is type ext4:

```
# mkfs.ext4 /dev/homevg/homelv
```

5. Add the new file system to the file system table and mount it:

```
# vi /etc/fstab
...
# grep home /etc/fstab
/dev/homevg/homelv      /home                   ext4    defaults        0 0
# mount /home
```

6. Create a sizable file on it with the **dd** command and show file system usage:

```
# dd if=/dev/zero of=/home/bigfile bs=1M count=500
500+0 records in
500+0 records out
524288000 bytes (524 MB) copied, 3.0718 s, 171 MB/s
# df -h | grep home
/dev/mapper/homevg-homelv
                      6.8G  644M  5.8G  10% /home
```

7. Show the volume group usage with the **vgdisplay** command:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  VG Size               6.88 GiB
  PE Size               4.00 MiB
  Total PE              1760
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       0 / 0
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

   This output shows that all physical extents in the volume group are used.

8. Add a second physical volume (that will be the target of the data move) to the volume group:

```
# vgextend homevg /dev/dasdf1
  Volume group "homevg" successfully extended
```

9. Show the volume group usage again:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  ...
  VG Size               13.75 GiB
  PE Size               4.00 MiB
  Total PE              3520
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

This output shows that the volume group doubled in size and now an equal number of free extents exist.

10. Move the data off the source physical volume with the **pvmove** command. The target does not need to be specified:

```
# pvmove /dev/dasde1
  /dev/dasde1: Moved: 0.0%
  /dev/dasde1: Moved: 8.0%
  /dev/dasde1: Moved: 18.9%
  /dev/dasde1: Moved: 34.2%
  /dev/dasde1: Moved: 49.1%
  /dev/dasde1: Moved: 63.2%
  /dev/dasde1: Moved: 77.6%
  /dev/dasde1: Moved: 92.7%
  /dev/dasde1: Moved: 100.0%
```

11. Show the volume group usage again:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  ...
  VG Size               13.75 GiB
  PE Size               4.00 MiB
  Total PE              3520
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

These free and used extents are the same; however, the data was moved.

12. Show the free and used extents on the source and target physical volumes with the **pvdisplay** command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
  --- Physical volume ---
  PV Name               /dev/dasde1
  VG Name               homevg
  PV Size               6.88 GiB / not usable 2.41 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              1760
```

```
Free PE               1760
Allocated PE          0
PV UUID               Jo2fa3-5cc0-y2Xs-e0DQ-wQXc-i3er-MPcckW

--- Physical volume ---
PV Name               /dev/dasdf1
VG Name               homevg
PV Size               6.88 GiB / not usable 2.41 MiB
Allocatable           yes (but full)
PE Size               4.00 MiB
Total PE              1760
Free PE               0
Allocated PE          1760
PV UUID               hme2qP-6ytn-Drg8-Wba4-rTU1-q1sV-pVZO3g
```

13. Remove the source physical volume:

```
# vgreduce homevg /dev/dasde1
  Removed "/dev/dasde1" from volume group "homevg"
```

The source volume is now ready for reassignment, or retirement.

Moving data from one physical volume to another physical volume without taking the file system offline was demonstrated.

14

Working with networks

This chapter describes the several miscellaneous tasks that you might need to perform and includes the following topics:

- “Setting up a private interconnect” on page 416
- “Creating a HiperSockets device between logical partitions” on page 418
- “Configuring a port group by using Link Aggregation Control Protocol” on page 421

14.1 Setting up a private interconnect

Having networked communications between different hosts that belong to a certain group can be beneficial. For example, certain legal databases must communicate to machines that scan documents for legal issues. Or, a web server and a certain back-end machine might need to communicate with each other without interference from other machines. Before live relocation, it was sufficient to merely set up a VSWITCH without an external interface to accomplish these tasks.

However, when you try to run this interconnect between hosts that run on a cross-central processor complex (CPC) SSI cluster, the private interconnect must be able to connect the network on the guests regardless of which CPC the guest is running on.

14.1.1 Directory Network Authorization

APAR VM65925 for z/VM 6.4, included in the base of z/VM 7.1 and later, provides a new and simplified process for defining and connecting guests to VSWITCHes called *Directory Network Authorization* (DNA). Instead of separate definitions for a virtual NIC and its authorization, DNA enhances the NICDEF directory statement to include authorization information.

Before DNA, a VSWITCH was defined for user-based (the default) or port-based access control. In user-based mode, a guest is permitted or denied to connect to a VSWITCH, and all ports that are connected to a VSWITCH feature the same access characteristics (such as ACCESS or TRUNK mode, and VLAN memberships). In port-based mode, a guest is granted access to connect to specific port numbers on a VSWITCH, and those ports might include different configured access characteristics.

When DNA is enabled, all VSWITCHes operate like the previous port-based mode. A port definition that corresponds to the access that us implied through the NICDEF directory entry is automatically defined for a guest when that virtual NIC connects to the VSWITCH.

DNA is the default operation mode for VSWITCHes on a system where the APAR is installed. Therefore, setting up an interconnect network can now be done by creating:

- A VSWITCH (the old way)
- A VLAN on an existing VSWITCH (the new “DNA-enabled” way)

14.1.2 Creating a VSWITCH for interconnect

An easy way to connect the network on the guests is to set up a virtual LAN (VLAN) for each of the required private interconnects on the external network. For each of these VLANs, then create a VLAN-aware VSWITCH with `PORTTYPE ACCESS`.

Complete the following steps:

1. Set up a network switch that connects to the mainframe and configure all necessary VLANs as tagged VLANs to the attached port.
2. Find a free port triplet on the Open Systems Adapter (OSA) device, for example, for the devices ***903 - 905***.
3. Edit the system configuration and add the following statement to the end of the file:

   ```
   DEFINE VSWITCH PRV01 RDEV 0903 ETH VLAN 75 PORTT ACCESS
   ```

4. Grant access to only the group of virtual machines that are on that network:

```
MODIFY VSWITCH PRV01 GRANT LINUXADM
MODIFY VSWITCH PRV01 GRANT LINUX5
```

5. Perform the same steps on all other members of the SSI.
6. Define a private Internet Protocol (IP) range for the group of hosts. It is a preferred practice to track the IP ranges and to not overlap them, even if the hosts do not connect to each other through a network.

14.1.3 Creating an interconnect VLAN on a VSWITCH

To use an existing VSWITCH for an interconnect VLAN, the VSWITCH must be VALN-aware. Consider the following points:

- If the VSWITCH is not yet VLAN-aware, it must be converted to being VLAN-aware before others VLANs can be added. Consult your network team about how this process is to be done. Some of the changes include the following examples:
  - The network ports your VSWITCH attaches to \ change from Access ports to Trunk ports
  - On your VSWITCH definition, you must specify the VLAN option and provide the Default VLAN ID. This is the VLAN ID to which the network team assigned all your VSWITCH traffic. When your VSWITCH becomes VLAN-aware, it becomes responsible for the default tagging instead of the network.
- If the VSWITCH is VLAN-aware, ask the network team to define a new VLAN to the network interfaces that are used by the VSWITCH. No definition is needed on the VSWITCH.

To attach the guests to the interconnect VLAN, they must have more virtual NICs defined. The extra NIC must have the VLAN keyword added, which specifies the VLAN ID of the interconnect VLAN. For example, the following example shows two NICs on a guest, one for normal network traffic and one for interconnect:

```
NICDEF 600 TYPE QDIO LAN SYSTEM MAINVSW
NICDEF 620 TYPE QDIO LAN SYSTEM MAINVSW VLAN 120
```

In this example, the VSWITCH was defined with VLAN DEFAULT 100, which assigns all network interfaces that are not specified with a VLAN ID to VLAN 100. The second interface specifies VLAN 120, which is the interconnect VLAN.

14.2 Creating a HiperSockets device between logical partitions

IBM HiperSockets devices can be used within a CEC to enable fast and secure connectivity between a Linux server and z/OS. The following actions are described:

- Verifying HiperSockets hardware definitions
- Creating a TCP/IP stack on z/OS
- Verifying HiperSockets hardware definitions
- Verifying connectivity

14.2.1 Verifying HiperSockets hardware definitions

Connectivity requires a HiperSockets IQD CHPID and devices that can be accessed by both the z/OS LPAR and the Linux z/VM LPAR. In Figure 2-3 on page 62, we defined a HiperSockets connection CHPID `F0` between z/OS LPAR A12 and z/VM LPAR A02 by using device `7000`.

This diagram is defined in the following input/output configuration program (IOCP) statements:

```
CHPID PATH=(CSS(0,1,2,3),F0),SHARED,                           *
      NOTPART=((CSS(0),(A04,A0C,A0D,A0E,A0F),(=)),(CSS(1),(A1B*
      ,A1D,A1E,A1F),(=)),(CSS(2),(A2E,A2F),(=)),(CSS(3),(A32,A*
      3D,A3E,A3F),(=))),TYPE=IQD
CNTLUNIT CUNUMBR=7000,                                         *
      PATH=((CSS(0),F0),(CSS(1),F0),(CSS(2),F0),(CSS(3),F0)), *
      UNIT=IQD
IODEVICE ADDRESS=(7000,16),UNITADD=00,CUNUMBR=(7000),UNIT=IQD
```

VM LPAR A02 and z/OS LPAR A12 can access the HiperSockets CHPID F0, and it is an IQD type.

14.2.2 Creating a TCP/IP stack on z/OS

To create a TCP/IP stack within z/OS to use the HiperSockets device, it is recommended to get assistance from your network team. For more information about HiperSockets connectivity, see *IBM HiperSockets Implementation Guide*, SG24-6816.

Complete the following steps to create a TCP/IP stack on z/OS:

1. Create a TCP/IP stack (which is called `TCPIPF` in this example) with a TCP/IP profile that uses the ***F0*** CHPID:

   ```
   VIEW       TCPIPF.SC42.TCPPARMS(TCPPROF) - 01.05
   Command ===>
   000085
   000086 DEVICE IUTIQDF0 MPCIPA
   000087 LINK   HIPERLF0    IPAQIDIO      IUTIQDF0
   000088
   ...
   000090 HOME
   000093    10.1.1.42     HIPERLF0
   ..
   000097 BEGINROUTES
   ..
   000102 ROUTE 10.1.1.0 255.255.255.0 = HIPERLF0 MTU 8192
   ```

```
000103 ENDROUTES
000104
000107 START IUTIQDFO
```

2. Put the CHPID identifier within the IUTIQDxx device statement. If it meets your sites' requirements, place the CHPID identifier in the `LINK` statements. Give the link a `HOME` address and `ROUTE` address according to your site networking requirements. Start your **`TCPIPF`** address space that uses this profile.
3. Issue the command **`D TCPIP,TCPIPF,NETSTAT,DEVL`** to verify the link information.

14.2.3 Configuring the HiperSockets interface on Linux

Complete the following steps to create a TCP/IP stack on Linux:

1. Request a free HiperSockets triplet from your system administrator.
2. Log on as `MAINT`, and verify the availability of the triplet:

```
===> q 7000-7002
OSA  7000 FREE    , OSA  7001 FREE    , OSA  7002 FREE
Ready; T=0.01/0.01 16:11:43
```

3. Attach the HiperSockets devices to the Linux image by using virtual device numbers. The command is issued from `Linux1` in this example:

```
===> attach 7004 LINUX2 E000
OSA 7000 ATTACHED TO LINUX1 E000
===> attach 7005 LINUX2 E001
OSA 7001 ATTACHED TO LINUX1 E001
===> attach 7003 LINUX E002
OSA 7002 ATTACHED TO LINUX1 E002
```

4. Verify the HiperSockets device type:

```
===> q 7003-7005
OSA  7003 ATTACHED TO LINUX2   E002 DEVTYPE HIPER      CHPID F0 IQD
OSA  7004 ATTACHED TO LINUX2   E000 DEVTYPE HIPER      CHPID F0 IQD
OSA  7005 ATTACHED TO LINUX2   E001 DEVTYPE HIPER      CHPID F0 IQD
```

5. Make the changes permanent with the following **`DIRM`** commands:

```
===> DIRM FOR LINUX1 DEDICATE E000 7004
===> DIRM FOR LINUX1 DEDICATE E001 7005
===> DIRM FOR LINUX1 DEDICATE E002 7003
```

Using Red Hat Enterprise Linux 7.1

Complete the following steps to create the `cio_ignore -r 0.0.e000,0.0.e001,0.0.e002` device:

1. From the Linux image, create a device group for the `E000` devices:

```
# echo 0.0.e000,0.0.e001,0.0.e002 > /sys/bus/ccwgroup/drivers/qeth/group
```

2. Bring the device online:

```
# echo 1 > /sys/devices/qeth/0.0.e000/online
```

3. Get the name of the devices from this command:

```
# cat /sys/devices/qeth/0.0.e000/if_name
   enccw0.0.e000
```

4. Create a network configuration file by using the **nmcli** command:

```
# nmcli con add type ethernet con-name hipersocket ifname enccw0.0.e000 ip4
10.0.0.1/21
# nmcli con mod hipersocket 802-3-ethernet.s390-nettype "qeth"
# nmcli con mod hipersocket 802-3-ethernet.s390-subchannels
"0.0.e000,0.0.e001,0.0.0e002"
# znetconf -A
```

5. Verify the `enccw0.0.e000` status with the **ip** and **lsqeth** command:

```
# ip addr show
...
3: enccw0.0.e000: <BROADCAST,MULTICAST,NOARP,UP,LOWER_UP> mtu 8192 qdisc
pfifo_fast state UNKNOWN qlen 1000
    link/ether 06:00:f0:09:00:03 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/21 brd 10.0.7.255 scope global enccw0.0.e000
       valid_lft forever preferred_lft forever
    inet6 fe80::400:f0ff:fe09:3/64 scope link
       valid_lft forever preferred_lft forever

# lsqeth
Device name                 : enccw0.0.e000
-------------------------------------------------------------------------
        card_type           : HiperSockets
        cdev0               : 0.0.e000
        cdev1               : 0.0.e001
        cdev2               : 0.0.e002
        chpid               : F0
        online              : 1
        portname            : no portname required
        portno              : 0
        route4              : no
        route6              : no
        state               : UP (LAN ONLINE)
        priority_queueing   : always queue 2
        fake_broadcast      : 0
        buffer_count        : 128
        layer2              : 0
        isolation           : none
        sniffer             : 0
```

SUSE Linux Enterprise Server

If you use the SUSE Linux Enterprise Server distribution of Linux, perform the following steps:

1. Configure the second network interface card (NIC) with the **qeth_configure** command:

```
# qeth_configure -t hsi 0.0.7000 0.0.7001 0.0.7002 1
```

2. Check whether the device was created:

```
# cat /proc/net/dev
```

3. If `hsi0` was created, you will see a file that is called `/etc/sysconfig/network/ifcfg-hsi0`. You will need to edit this file by using the following command:

```
# vi /etc/sysconfig/network/ifcfg-hsi0
BOOTPROTO='static'
IPADDR='10.1.1.46/24'
STARTMODE='onboot'
```

```
NAME='HIPERSOCKETS (0.0.7400)'
```

4. Start the `hsi0` device with the **`ifup`** command:

```
# ifup hsi0
```

5. Check the status of the interface:

```
# ip a s hsi0
```

The HiperSockets device is now up.

14.2.4 Verifying connectivity

To verify that the HiperSockets device is functioning, perform the following steps:

1. Ping from z/OS UNIX Systems Services:

```
USER1 @ SC42:/u/user1>ping 10.1.1.43
CS V1R13: Pinging host 10.1.1.43
Ping #1 response took 0.000 seconds.
```

2. Ping from the Red Hat Enterprise Linux that runs on `ITSOZVM1`:

```
[root@virtcook1 etc]# ping 10.1.1.42
PING 10.1.1.42 (10.1.1.42) 56(84) bytes of data.
64 bytes from 10.1.1.42: icmp_seq=1 ttl=64 time=0.025 ms
```

3. Ping from the SUSE Linux Enterprise Server that runs on `ITSOZVM2`:

```
linuxadmin:/etc/sysconfig/network # ping 10.1.1.46
PING 10.1.1.46 (10.1.1.46) 56(84) bytes of data.
```

This process shows that the HiperSockets device is working.

14.3 Configuring a port group by using Link Aggregation Control Protocol

To aggregate multiple OSA-Express ports, *port groups* can be defined on z/VM and attached to a virtual switch. Traffic is distributed over the multiple ports automatically.

The initial support for VSwitch Link Aggregation required that the OSA ports were used only by a single z/VM VSWITCH and were not shared with another system (even a Linux guest in the same LPAR attaching to the OSA directly).

Later, the Multi-VSwitch Link Aggregation capability was introduced. This support, also known as *Global VSwitch*, allows a port group to be defined as shared between z/VM VSwitches, even across multiple LPARs.

14.3.1 Exclusive-mode port group

Connectivity by using a port group in Exclusive Mode requires OSA devices that are used by only one z/VM LPAR. This example uses four-port OSA express cards, which use two ports for each CHPID (see Figure 14-1 on page 422).

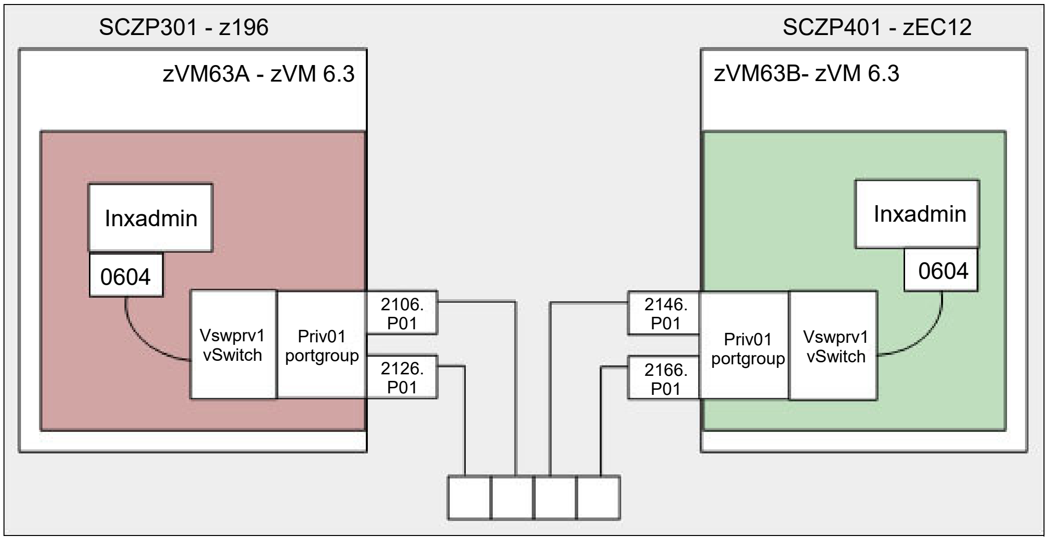


Figure 14-1 Port group Priv01 connectivity

Note: Port number 1, not port number 0, was used for this connection.

`ITSOZVM1` port group `priv1` has the following details:

```
CHPID 00 portnumber 1 OSA device 2106
CHPID 01 portnumber 1 OSA device 2126
```

`ITSOZVM2` port group `priv1` has the following details:

```
CHPID 00 portnumber 1 OSA device 2046
CHPID 01 portnumber 1 OSA device 2066
```

Use the following steps to accomplish this task:

1. Create the port group on the first SSI member (`ITSOZVM1` in this example):

   ```
   ===> set port group priv01 join 2106.p01 2126.p01
   ```

2. Create the port group on the second SSI member (`ITSOZVM2` in this example):

   ```
   ===> set port group priv01 join 2046.p01 2066.p01
   ```

   Note: Consider the following points:

   - Link Aggregation Control Protocol (LACP) is set as active, by default. To use LACP, the network switch needs LACP to be active on the ports to which the CHPIDs connect. If LACP is not active in the switch, the port group does not activate when it is defined to a VSWITCH.
   - If you receive message `HCPSWU2832E`, the LPAR does not have exclusive use of the device. Another LPAR has the device online. However, exclusive use does not require that only one LPAR in the I/O configuration data set (IOCDS) has the CHPID defined and the devices dedicated.

3. Define the virtual switch by using the `priv01` port group on all members of the SSI (`ITSOZVM1` and `ITSOZVM2` in this example):

   ```
   ===> define vswitch vswprv1 rdev none ethernet vlan aware group priv01 gvrp
   ```

14.3.2 Multiple VSWITCH Link Aggregation

As you can see from the previous example, exclusive-mode link aggregation does not scale over many z/VM systems. The Global VSwitch capability relieved this limitation by allowing link aggregation OSA ports to be shared between z/VM LPARs.

To allow exclusive-mode link aggregation to scale over many z/VM systems, an IVL Domain is a group of up to 16 z/VM systems that can share link aggregation ports. Connectivity for IVL Domains is implemented over an Ethernet-based IVL Network, which is defined by using IVL VSwitches and OSA ports.

Up to 16 z/VM systems can be part of an IVL Domain, and an IVL Network can support many IVL Domains. Up to eight Domains are supported without the use of VLANs on the IVL Network; up to eight Domains per IVL VLAN are supported with VLANs on the IVL Network).

The ports used for the IVL Network must be separate from the link aggregation ports. However, because the IVL Network does not carry much traffic, OSA ports that are used for non-link-aggregated connections can be used for the IVL Network. However, it is highly recommended that the IVL Network be a separate VLAN from other network traffic.

The process that is used to create a Global VSwitch includes the overall tasks:

- Allocate ports for use by the IVL Network and the Global VSwitch.
- Define the IVL VSwitch on each of the z/VM systems (making sure to use the suitable Domain and VLAN definitions as required).
- Verify connectivity across the IVL Network.
- Define the Port Group by using the **LACP ACTive SHAred** option.
- Define the VSwitch by using the **GLOBAL** option.

Each of these steps is described next.

Allocating ports for the IVL Network and Global VSwitch

Decide which available ports are to be used for the IVL Network and the Global VSwitch. We recommend that the Global VSwitch be defined across the fastest cards that are available in your environment. Also, although it is not technically a requirement, we recommend that all the cards that are used in the Global VSwitch be the same type (or at least the same link speed).

Because the traffic requirement of the IVL Network is modest, OSA-Express 1000BaseT or Gigabit Ethernet cards can be used (even if the Global VSwitch uses 10 Gigabit cards).

Defining the IVL VSwitch on each system

The IVL VSwitch is defined by using the **DEFINE VSWITCH** command, as with a normal VSwitch, but the type of **IVL** must be used. Example 14-1 shows the result of creating the IVL VSwitch on the first of our systems.

Example 14-1 Creating the IVL VSwitch on the first system

```
define vswitch ivl type ivl rdev 1913.p01 1933.p01
VSWITCH SYSTEM IVL is created
HCPIVA3164I System RDBKZVMF is connected to IVL Domain A through VSWITCH IVL wit
h MAC address 02-00-00-00-00-00.
HCPSWU3230I Priority Queuing is not enabled for device 1913.P01 for VSWITCH SYST
EM IVL.
HCPSWU3230I Priority Queuing is not enabled for device 1933.P01 for VSWITCH SYST
```

```
EM IVL.
Ready; T=0.01/0.01 03:38:21
HCPSWU2830I VSWITCH SYSTEM IVL status is ready.
HCPSWU2830I DTCVSW3 is VSWITCH controller for device 1913.P01.
HCPSWU3181I The system is initiating IVL Member Discovery on IVL Domain A.
HCPSWU2830I DTCVSW4 is VSWITCH controller for backup device 1933.P01.
```

In our example, we used the default domain "A" and no VLAN support in the IVL VSwitch (our network switch provided VLAN isolation for us).

On the second system, we repeated the operation with the values for RDEV reversed. This configuration improves resilience by having each system use a different OSA port for its primary IVL Network connection. Example 14-2 shows the creation of the IVL VSwitch on the second system.

Example 14-2 Creating the IVL VSwitch on the second system

```
define vswitch ivl type ivl rdev 1933.p01 1913.p01
VSWITCH SYSTEM IVL is created
HCPIVA3164I System RDBKZVMC is connected to IVL Domain A through VSWITCH IVL wit
h MAC address 02-03-11-00-00-00.
Ready; T=0.01/0.01 05:28:07
HCPSWU2830I VSWITCH SYSTEM IVL status is ready.
HCPSWU2830I DTCVSW3 is VSWITCH controller for device 1933.P01.
HCPSWU3181I The system is initiating IVL Member Discovery on IVL Domain A.
HCPSWU3175I System RDBKZVMF has been added to IVL Domain A.
HCPSWU2830I DTCVSW4 is VSWITCH controller for backup device 1913.P01.
```

We also saw a message on the first system console to indicate that the second system was discovered by the first:

```
HCPSWU3175I System RDBKZVMC has been added to IVL Domain A.
```

Verifying IVL connectivity

We used the **QUERY VSWITCH** command to check the status of our IVL Network. Example 14-3 shows the result of this query.

Example 14-3 QUERY VSWITCH on the IVL VSwitch

```
q vswitch ivl
VSWITCH SYSTEM IVL        Type: IVL     Domain: A     Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  PORTBASED LOCAL
  VLAN Unaware
  MAC address: 02-00-00-00-00-0E    MAC Protection: Unspecified
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Ready                         PriQueuing: FORCED OFF
  PMTUD setting: EXTERNAL   PMTUD value: 9000     Trace Pages: 8
  RDEV: 1913.P01 VDEV: 0609 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.0174
  RDEV: 1933.P01 VDEV: 0603 Controller: DTCVSW4  BACKUP
        Adapter ID: 3907000BB4B7.010C
 IVL Port:
    Adapter Owner: SYSTEM   NIC: FFFD.P00 Name: UNASSIGNED  Type: QDIO
```

```
RX Packets: 1132       Discarded: 1062       Errors: 0
TX Packets: 1388       Discarded: 0          Errors: 0
RX Bytes: 52312                TX Bytes: 152704
Device: FFFD  Unit: 000   Role: DATA       Port: 2100
Options: Ethernet Broadcast
  Unicast MAC Addresses:
    02-00-00-00-00-00
  Multicast MAC Addresses:
    03-FF-FF-FF-FF-01
```

This display shows the “IVL Port” section, which confirms it is an IVL VSwitch. The Multicast MAC Address that is listed ends in 01, which is the address that is assigned to IVL Domain A.

More information about the status of the IVL is shown by running the **QUERY VMLAN** command. Example 14-4 shows the VMLAN details from our first system.

Example 14-4 QUERY VMLAN command to show IVL details

```
q vmlan
VMLAN maintenance level:
  Latest Service: Base
VMLAN MAC address assignment:
  System MAC Protection: OFF
  MACADDR Prefix: 020000 USER Prefix: 020000
  MACIDRANGE SYSTEM: 000001-FFFFFF
             USER:   000000-000000
VMLAN default accounting status:
  SYSTEM Accounting: OFF       USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 3
  TRANSIENT  Limit: INFINITE   Current: 0
Trace Pages: 8
VMLAN Directory Network Authorization: ENABLED

IVL Domain: A   MAC address: 03-FF-FF-FF-FF-01  VLAN: <none>
IVL Domain Heartbeat Timeout: 30
IVL Domain Capability: C000000000000000
   Member: RDBKZVMF MAC address: 02-00-00-00-00-00
     State: Active
     Heartbeat Count: <local>
     Member Capability: C000000000000000 Maintenance Level: V720

   Member: RDBKZVMC MAC address: 02-03-11-00-00-00
     State: Active
     Heartbeat Count: 6
     Member Capability: C000000000000000 Maintenance Level: V710
```

Connectivity to the members of the IVL Domain can be verified by using the **SET VSWITCH IVL IVLPORT** command. Example 14-5 shows the result when we tested connectivity in our test setup.

Example 14-5 SET VSWITCH IVL IVLPORT command

```
set vswitch ivl ivlport ping all
Command complete
HCPIVA3187I Pinging MAC 03-FF-FF-FF-FF-01 : bytes=8096 of data.
Ready; T=0.01/0.01 06:47:32
HCPIVA3187I Reply from MAC 02-03-11-00-00-00 : bytes=8096 time=0630 us
```

Defining the Shared Port Group

After the IVL Network was verified, we then defined the port group. Through the IVL Domain, the port group is created on the second system when it is defined on the first system.

Example 14-6 shows the commands that we issued on the first system to create the shared port group. First, we defined the group as a shared port group; then, we added the two OSA ports to the group.

Example 14-6 Using SET PORT GROUP commands to define a shared port group

```
set port group grpglbl lacp active shared
Port group GRPGLBL is created
Ready; T=0.01/0.01 07:05:26
set port group grpglbl join 1910 1930
Port group GRPGLBL is updated
Ready;
```

To verify that the IVL Domain was functioning correctly, we ran a **QUERY PORT GROUP** command on the second system. When running the **QUERY PORT GROUP** command without parameters, we saw the response `No active groups found`. This message is expected because the port group does not become active until it is connected to a VSwitch.

Displaying inactive port groups can be done by using the **ALL INActive** parameter on the **QUERY PORT GROUP** command, or by displaying the group by name.

We used the display-by-name method to show the group. As shown in Example 14-7, we saw that the group was defined on the second system automatically through the IVL.

Example 14-7 QUERY PORT GROUP command

```
q port group grpglbl
  Group: GRPGLBL.0  Inactive LACP Mode: Active   Shared
  VSWITCH <none>                            ifIndex:  2112
  Load Balancing: Collaborative        Interval: 300
  RDEV: 1930.P00   Adapter ID: 3907000BB4B7.010C
  RDEV: 1910.P00   Adapter ID: 3907000BB4B7.0174
Ready;
```

We saw the same output when we issued the command on the first system.

Defining the Global VSwitch

A global port group can be defined in LACP mode only. We had to coordinate with our network management team to have the correct configuration made in our network switch (see 14.3.4, “Link Aggregation Control Protocol” on page 434 for more information about what can occur if the switch configuration is incorrect). After the switch setup was complete, we created the Global VSwitch on one of the systems (RCBKZVMC), as shown in Example 14-8.

Example 14-8 Creating a Global VSwitch on one system

```
define vswitch vswglbl ethernet group grpglbl global
VSWITCH SYSTEM VSWGLBL is created
Ready;
HCPSWU3165I Device 1930.P00 for VSWITCH VSWGLBL is now the Active LAG Port Contr
oller for shared port group GRPGLBL.
HCPSWU2830I VSWITCH SYSTEM VSWGLBL status is ready.
HCPSWU2830I DTCVSW1 is VSWITCH controller for device 1930.P00.
HCPSWU2855I Device 1930.P00 for VSWITCH VSWGLBL is enabled for port group GRPGLB
L by LACP.
HCPSWU3165I Device 1910.P00 for VSWITCH VSWGLBL is now the Active LAG Port Contr
oller for shared port group GRPGLBL.
HCPSWU2830I DTCVSW3 is VSWITCH controller for device 1910.P00.
HCPSWU2855I Device 1910.P00 for VSWITCH VSWGLBL is enabled for port group GRPGLB
L by LACP.
```

Over the course of a few seconds, the LACP negotiation starts with the network switch over the first port in the port group. This process is indicated by the first pair of **HCPSWU2830I** and **HCPSWU2855I** messages, followed by HCPSWU3165I when the LACP negotiation completes. Later (in our case it was about 20 - 30 seconds) the second port in the group is added and the second pair of **HCPSWU2830I** and **HCPSWU2855I** messages appears.

We performed a display of the new VSwitch by using **QUERY VSWITCH DETAILS**. Example 14-9 shows the output of this command.

Example 14-9 QUERY VSWITCH DETAILS on the Global VSwitch

```
q vswitch vswglbl details
VSWITCH RDBKZVMC.VSWGLBL  Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED GLOBAL
  VLAN Unaware
  MAC address: 02-03-11-00-00-07    MAC Protection: Unspecified
  IPTimeout: 5        QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Ready                         PriQueuing: OFF
  PMTUD setting: EXTERNAL   PMTUD value: 9000     Trace Pages: 8
  Group: GRPGLBL.0  Active   LACP Mode: Active   Shared
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW1  ACTIVE
        Adapter ID: 3907000BB4B7.010C
    Uplink Port Connection:
      MAC address: 02-03-11-00-00-08
      RX Packets: 82        Discarded: 0         Errors: 0
      TX Packets: 114       Discarded: 0         Errors: 0
      RX Bytes: 10168               TX Bytes: 14136
      Device: 0603  Unit: 000   Role: DATA       Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
```

```
      LAG Port Controller: Active
  RDEV: 1910.P00 VDEV: 0603 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.0174
    Uplink Port Connection:
      MAC address: 02-03-11-00-00-09
      RX Packets: 82        Discarded: 0         Errors: 0
      TX Packets: 106       Discarded: 0         Errors: 0
      RX Bytes: 10168               TX Bytes: 13144
      Device: 0603  Unit: 000   Role: DATA      Port: 2050
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Active
 Member:  RDBKZVMC.VSWGLBL
    State: Synchronized
Ready;
```

The VSwitch details display shows some information about the port group. The members of the Global VSwitch at the bottom of the display. Because only RDBKZVMC (the first system) has the Global VSwitch defined, it is the only system listed.

We ran a display of the port group to see if any other information was displayed there. The output of **QUERY PORT GROUP VSWGLBL** was much the same as we saw in the VSwitch display, but adding **DETAILS** to the query displayed much LACP information (see Example 14-10).

Example 14-10 QUERY PORT GROUP DETAILS output

```
q port group grpglbl details
  Group: GRPGLBL.0  Active   LACP Mode: Active   Shared
  VSWITCH RDBKZVMC.VSWGLBL                  ifIndex:  2112
  Load Balancing: Collaborative         Interval: 300
  GROUP Information:
    PORT Information - Total Frames per Interval:
                          Local       Local                      Total
      Device  Status      Previous      Current                 Previous
      ------  ------- ------------ ------------  -----------------------
       1930   Active           10           10                        10
       1910   Active           16            6                        16
      ----------------------------------------------------------------
      Total Port Group Frames:                                        26

      Last Load Balance: RDBKZVMC  Date: 10/18/20   Time: 02:59:37

    ROUTING Information - Frame Distribution per Interval:
      MAC     Device      Previous      Current
      ----- --------- ------------ ------------
       0      1930             0            0
       1      1910             0            0
       2      1930             0            0
       3      1910             0            0
       4      1930             0            0
       5      1910             0            0
       6      1930             0            0
       7      1910             0            0
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW1  ACTIVE
        Adapter ID: 3907000BB4B7.010C
    Uplink Port Connection:
```

```
      MAC address: 02-03-11-00-00-08
      RX Packets: 1153      Discarded: 0         Errors: 0
      TX Packets: 1562      Discarded: 0         Errors: 0
      RX Bytes: 142972              TX Bytes: 193688
      Device: 0603  Unit: 000   Role: DATA      Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Active
    PROTOCOL Counters:
      LACP RX: 1149         Marker RX: 0
      LACP TX: 1558         Marker TX: 4         Timeouts: 4
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07       Oper  Key: 2
      Port Priority: 32768   Port: 2049        Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
    PARTNER Information:
      System ID: 32768,A4-8C-DB-94-58-00       Oper  Key: 101
      Port Priority: 32768   Port: 9           Group Key: 101
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
  RDEV: 1910.P00 VDEV: 0603 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.0174
    Uplink Port Connection:
      MAC address: 02-03-11-00-00-09
      RX Packets: 1147      Discarded: 0         Errors: 0
      TX Packets: 1554      Discarded: 0         Errors: 0
      RX Bytes: 142228              TX Bytes: 192696
      Device: 0603  Unit: 000   Role: DATA      Port: 2050
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Active
    PROTOCOL Counters:
      LACP RX: 1147         Marker RX: 0
      LACP TX: 1554         Marker TX: 0         Timeouts: 0
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07       Oper  Key: 2
      Port Priority: 32768   Port: 2050        Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
    PARTNER Information:
      System ID: 32768,A4-8C-DB-94-58-00       Oper  Key: 101
      Port Priority: 32768   Port: 7           Group Key: 101
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
 Member:  RDBKZVMC
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Connected
 Member:  RDBKZVMF
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Inactive
Ready;
```

At the end of this details display, we can see that the status RDBKZVMC (the system we created the VSwitch on) is `Connected`, while the status of RDBKZVMF (the second system) is `Inactive`. This result occurs because the Global VSwitch is not yet created on the second system.

On the second system, we checked its view of the port group to make sure it was active and ready for use. We then issued the command to define the Global VSwitch on RDBKZVMF. These commands and their output are shown in Example 14-11.

Example 14-11 Setting up the Global VSwitch on the second system

```
q port group grpglbl
  Group: GRPGLBL.0  Inactive LACP Mode: Active   Shared
  VSWITCH <none>                             ifIndex:  2112
  Load Balancing: Collaborative         Interval: 300
  RDEV: 1930.P00   Adapter ID: 3907000BB4B7.010C
  RDEV: 1910.P00   Adapter ID: 3907000BB4B7.0174
Ready;
define vswitch vswglbl ethernet group grpglbl global
VSWITCH SYSTEM VSWGLBL is created
Ready;
HCPSWU2830I VSWITCH SYSTEM VSWGLBL status is ready.
HCPSWU2830I DTCVSW3 is VSWITCH controller for device 1930.P00.
HCPSWU2855I Device 1930.P00 for VSWITCH VSWGLBL is enabled for port group GRPGLB
L by LACP.
HCPSWU2830I DTCVSW4 is VSWITCH controller for device 1910.P00.
HCPSWU2855I Device 1910.P00 for VSWITCH VSWGLBL is enabled for port group GRPGLB
L by LACP.
```

As we saw with the first system, the VSwitch is defined and becomes active after a short period. We issued **Q VSWITCH DETAILS** on RDBKZVMF. The results are shown in Example 14-12.

Example 14-12 QUERY VSWITCH DETAILS for the Global VSwitch on the second system

```
q vswitch vswglbl details
VSWITCH RDBKZVMF.VSWGLBL  Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED GLOBAL
  VLAN Unaware
  MAC address: 02-06-50-00-00-0B    MAC Protection: Unspecified
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Ready                         PriQueuing: OFF
  PMTUD setting: EXTERNAL   PMTUD value: 9000     Trace Pages: 8
  Group: GRPGLBL.0  Active   LACP Mode: Active   Shared
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.010C
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0C
      RX Packets: 50        Discarded: 0         Errors: 0
      TX Packets: 4         Discarded: 0         Errors: 0
      RX Bytes: 6200                TX Bytes: 496
      Device: 0603  Unit: 000   Role: DATA      Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Standby
  RDEV: 1910.P00 VDEV: 0603 Controller: DTCVSW4  ACTIVE
        Adapter ID: 3907000BB4B7.0174
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0D
```

```
      RX Packets: 48        Discarded: 0         Errors: 0
      TX Packets: 0         Discarded: 0         Errors: 0
      RX Bytes: 5952                TX Bytes: 0
      Device: 0603  Unit: 000   Role: DATA       Port: 2050
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Standby
 Member:  RDBKZVMF.VSWGLBL
    State: Synchronized
 Member:  RDBKZVMC.VSWGLBL
    State: Synchronized
Ready;
```

We see the following important details in this display:

- The MAC addresses against each of the ports are different on each system. This result shows that although the Global VSwitch functions like a logical single VSwitch that is present in both systems, it is two separate VSwitches that share the global port group. For this reason, the VSwitch must be defined on each system, while the port group did not need to be defined.
- Now that the Global VSwtich is defined on our second system, RDBKZVMF now appears as `Synchronized` at the end of the **QUERY VSWITCH DETAILS** output.

For completeness, we repeated the **Q PORT GROUP DETAILS** display on RDBKZVMF to compare the output (see Example 14-13).

Example 14-13 QUERY PORT GROUP DETAILS display on the second system

```
q port group grpglbl details
  Group: GRPGLBL.0  Active   LACP Mode: Active   Shared
  VSWITCH RDBKZVMF.VSWGLBL                  ifIndex:  2112
  Load Balancing: Collaborative         Interval: 300
  GROUP Information:
    PORT Information - Total Frames per Interval:
                          Local        Local                     Total
      Device  Status      Previous     Current                Previous
      ------  ------- ------------ ------------  -----------------------
       1930   Active            32           11                       42
       1910   Active            27           17                       37
      ------------------------------------------------------------------
      Total Port Group Frames:                                        79

      Last Load Balance: RDBKZVMF  Date: 10/18/20   Time: 08:12:57

    ROUTING Information - Frame Distribution per Interval:
      MAC     Device      Previous      Current
      ----- --------- ------------ ------------
       0      1930             0            0
       1      1910             0            0
       2      1930             0            0
       3      1910             0            0
       4      1930             0            0
       5      1910             0            0
       6      1930             0            0
       7      1910             0            0
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.010C
```

```
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0C
      RX Packets: 1765      Discarded: 0         Errors: 0
      TX Packets: 4         Discarded: 0         Errors: 0
      RX Bytes: 218860              TX Bytes: 496
      Device: 0603  Unit: 000   Role: DATA      Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Standby
    PROTOCOL Counters:
      LACP RX: 1765         Marker RX: 0
      LACP TX: 0            Marker TX: 4         Timeouts: 4
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07       Oper  Key: 2
      Port Priority: 32768   Port: 2049        Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
    PARTNER Information:
      System ID: 32768,A4-8C-DB-94-58-00       Oper  Key: 101
      Port Priority: 32768   Port: 9           Group Key: 101
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
  RDEV: 1910.P00 VDEV: 0603 Controller: DTCVSW4  ACTIVE
        Adapter ID: 3907000BB4B7.0174
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0D
      RX Packets: 1767      Discarded: 0         Errors: 0
      TX Packets: 0         Discarded: 0         Errors: 0
      RX Bytes: 219108              TX Bytes: 0
      Device: 0603  Unit: 000   Role: DATA      Port: 2050
      Partner Switch Capabilities: No_Reflective_Relay
      LAG Port Controller: Standby
    PROTOCOL Counters:
      LACP RX: 1767         Marker RX: 0
      LACP TX: 0            Marker TX: 0         Timeouts: 0
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07       Oper  Key: 2
      Port Priority: 32768   Port: 2050        Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
    PARTNER Information:
      System ID: 32768,A4-8C-DB-94-58-00       Oper  Key: 101
      Port Priority: 32768   Port: 7           Group Key: 101
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
 Member:  RDBKZVMF
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Connected
 Member:  RDBKZVMC
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Connected
Ready;
```

At the bottom of the display we can see that both systems that are defined to the port group are now showing a status of `Connected`, which is what we expect now that the Global VSwitch is defined on both systems.

Looking closer, we can see that although the MAC address detail for each port corresponds to RDBKZVMF, the MAC addresses that are shown in the LACP ACTOR details areas are the MAC addresses of RDBKZVMC. This result shows that the first system to define the Global VSwitch becomes the system that manages the LACP protocol communication with the network switch.

14.3.3 Global VSwitch recovery

We tested what occurs if RDBKZVMC was removed from the group. We issued the **SET VSWITCH VSWGLBL DISCONN** command on RDBKZVMC to disconnect it from the port group. Example 14-14 shows the messages that appeared on RDBKZVMF when we issued the disconnect command, and the output of a port group detail display (with some lines removed).

Example 14-14 Global VSwitch recovery, and status display

```
HCPSWU3165I Device 1930.P00 for VSWITCH VSWGLBL is now the Active LAG Port Contr
oller for shared port group GRPGLBL.
HCPSWU3165I Device 1910.P00 for VSWITCH VSWGLBL is now the Active LAG Port Contr
oller for shared port group GRPGLBL.
q port group grpglbl details
  Group: GRPGLBL.0  Active   LACP Mode: Active   Shared
  VSWITCH RDBKZVMF.VSWGLBL                  ifIndex:  2112
  Load Balancing: Collaborative         Interval: 300
. . .
      Last Load Balance: RDBKZVMF  Date: 10/18/20   Time: 08:32:57
. . .
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW3  ACTIVE
        Adapter ID: 3907000BB4B7.010C
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0C
. . .
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07        Oper  Key: 2
      Port Priority: 32768   Port: 2049         Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
. . .
  RDEV: 1910.P00 VDEV: 0603 Controller: DTCVSW4  ACTIVE
        Adapter ID: 3907000BB4B7.0174
    Uplink Port Connection:
      MAC address: 02-06-50-00-00-0D
. . .
    ACTOR Information:
      System ID: 32768,02-03-11-00-00-07        Oper  Key: 2
      Port Priority: 32768   Port: 2050         Group Key: 2
      State: 3D - LACP_Active   Slow AGG SYNC DIST COLL
. . .
 Member:  RDBKZVMF
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Connected
 Member:  RDBKZVMC
      Scope: Synchronized
      LAG Synchronization token: 0000000000000000
      Mode:  Inactive
Ready;
```

Again we can see at the bottom of the display that the status of RDBKZVMC is `Inactive` as a result of being disconnected from the group. We can also see that at the time the disconnection of RDBKZVMC occurred, RDBKZVMF performed a load balance of the LACP protocol.

Also, the MAC addresses in the ACTOR fields did not change, even though RDBKZVMC is no longer attached to the VSwitch. Instead of changing the MAC addresses, which is disruptive to LACP, the new controlling system (in this case RDBKZVMF) takes over the MAC addresses of RDBKZVMC's VSwitch to allow LACP to continue. This process happens internally in the port group because the MAC addresses do not appear in a **`QUERY VSWITCH DETAILS`** listing.

14.3.4 Link Aggregation Control Protocol

When setting up VSwitch Link Aggregation, it is highly recommended that the network switch definition is set up for Link Aggregation Control Protocol (LACP) operation. As defined in IEEE 802.1AX (formerly 802.3ad), LACP is the default operating mode of a z/VM port group. LACP is required when a port group is shared in Multi-VSwitch LAG configuration.

Various techniques are available for link aggregation in the networking world. Terms, such as *Etherchannel* and *port trunk* are used to describe different techniques that are used by various switch and operating system vendors to describe their link aggregation methods. These techniques are often proprietary and incompatible with each other.

Example 14-15 shows an example where a port group (in the example, it is an exclusive port group, but the same situation exists for a shared port group) is not functioning because LACP is unavailable. This issue occurs is because the network switch does not support LACP, or the switch ports the z/VM port group is attached to are not configured for LACP.

Example 14-15 Unsuccessful port group activation example

```
define vswitch vswglbl ether group glblgrp
VSWITCH SYSTEM VSWGLBL is created
Ready; T=0.01/0.01 02:00:17
HCPSWU2830I VSWITCH SYSTEM VSWGLBL status is ready.
HCPSWU2830I DTCVSW1 is VSWITCH controller for device 1910.P00.
HCPSWU2855I Device 1910.P00 for VSWITCH VSWGLBL is disabled for port group GLBLG
RP by LACP.
HCPSWU2830I DTCVSW2 is VSWITCH controller for device 1930.P00.
HCPSWU2855I Device 1930.P00 for VSWITCH VSWGLBL is disabled for port group GLBLG
RP by LACP.
q port group glblgrp
  Group: GLBLGRP    Active   LACP Mode: Active   Exclusive
  VSWITCH SYSTEM VSWGLBL                     ifIndex:  2112
  Load Balancing: Independent           Interval: 300
  RDEV: 1910.P00 VDEV: 0606 Controller: DTCVSW1  ACTIVE
        Adapter ID: 3907000BB4B7.0174
        Status: Error       Reason: LACP not enabled on partner
    Uplink Port Connection:
      MAC address: 02-00-00-00-00-0B
      RX Packets: 0         Discarded: 2         Errors: 2
      TX Packets: 70        Discarded: 0         Errors: 0
      RX Bytes: 0                   TX Bytes: 8680
      Device: 0606  Unit: 000   Role: DATA      Port: 2049
      Partner Switch Capabilities: No_Reflective_Relay
  RDEV: 1930.P00 VDEV: 0603 Controller: DTCVSW2  ACTIVE
        Adapter ID: 3907000BB4B7.010C
```

```
      Status: Error       Reason: LACP not enabled on partner
    Uplink Port Connection:
      MAC address: 02-00-00-00-00-0C
      RX Packets: 0          Discarded: 3          Errors: 1
      TX Packets: 66         Discarded: 0          Errors: 0
      RX Bytes: 0                    TX Bytes: 8184
      Device: 0603  Unit: 000   Role: DATA       Port: 2050
      Partner Switch Capabilities: No_Reflective_Relay
Ready; T=0.01/0.01 02:03:37
```

The port group **GLBLGRP** was defined with LACP, and the VSwitch **VSWGLBL** was then defined by using the group. The error reason "LACP not enabled on partner" is shown against both of the OSA ports in the group.

Only an exclusive-mode port group can operate without LACP, and in that configuration, traffic for the group is distributed between the ports in the group without being managed by a control protocol. Depending on the operating characteristics of the network switch, this configuration might result in traffic not being handled optimally. Some modern network equipment can operate poorly and even experience traffic loss when traffic appears across different ports unexpectedly.

Red Hat Enterprise Linux

If you are on a Red Hat Enterprise Linux system, perform the following steps to create the network device ETH1:

1. From the Linux image, create a device group for the 0604 devices:

   ```
   # echo 0.0.0604,0.0.0605,0.0.0606 > /sys/bus/ccwgroup/drivers/qeth/group
   ```

2. Bring the device online with the following command:

   ```
   # echo 1 > /sys/devices/qeth/0.0.0604/online
   ```

3. Get the name of the device:

   ```
   # cat /sys/devices/qeth/0.0.0604/if_name
   eth1
   ```

4. Create a network configuration file by using the name eth1 in the file: /etc/sysconfig/network-scripts/ifcfg-eth1:

   ```
   ===> vi /etc/sysconfig/network-scripts/ifcfg-eth1
   #IBM QETH
   DEVICE=eth1
   BOOTPROTO=static
   IPADDR=10.1.1.47
   NETMASK=255.255.255.0
   NETTYPE=qeth
   ONBOOT=yes
   SUBCHANNELS=0.0.0604,0.0.0605,0.0.0606
   TYPE=ethernet
   ARP=no
   ```

5. Start the eth1 network device with the **ifup** command:

   ```
   ===> ifup eth1
   ```

6. Verify the status of eth1 with the **ifconfig** command:

```
===> ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 02:00:0B:00:00:0B
          inet addr:10.1.1.47  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::bff:fe00:b/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2464 (2.4 KiB)  TX bytes:350 (350.0 b)
```

SUSE Linux Enterprise Server 12

If you are on a SUSE Linux Enterprise Server system, perform the following steps to create the network device ETH1:

1. Run the following command to create the device on LNXADMIN:

```
# qeth_configure -l -t qeth 0.0.0604 0.0.0605 0.0.0606 1
```

2. Create the interface eth1 by using the file /etc/sysconfig/network/ifcfg-eth1:

```
# vi /etc/sysconfig/network/ifcfg-eth1
BOOTPROTO='static'
IPADDR='10.1.1.48/24'
STARTMODE='onboot'
NAME='OSA Express(0.0.0604)'
```

3. Open the eth1 device with the **ifup** command:

```
# ifup eth1
```

4. Test the connectivity between each Linux image.

You now have a functioning network interface that uses port groups.

14.4 Linux network commands

Regardless of which Linux distribution you are using, some commands are available for interacting with network functions.

Users who worked with Linux (or UNIX) for several years might be used to older commands that are bundled with the `net-tools` package, most of which are deprecated.

The `iproute2` suite replaced the net-tools package. If you are still relying on the deprecated commands, begin moving to the new `iproute2` version as soon as you can. Some commonly used commands are listed in Table 14-1. Many of the modern commands can be abbreviated; the full command is shown with parenthesis around the optional full-length version.

Table 14-1 Linux networking commands

Obsolete commands	Modern commands	Purpose
`arp`	**`ip n(eighbor)`**	Neighboring IP information
`ifconfig`	**`ip a(ddress)`** **`ip l(ink)`** **`ip -stats`**	IP addressing values and stats
`ipmaddr`	**`ip m(addr)`**	Multicast values and status
`iptunnel`	**`ip t(unnel)`**	Set and review IP tunnel values
`netstat` `route`	**`ip r(oute)`** **`ss`**	IP routing and state information
`nameif`	**`ip link`** **`ifrename`**	Rename an interface
`mii-tool`	**`ethtool`**	Work with Ethernet values
`iwconfig`	**`iw`**	Work with WiFi NICs
`route`	**`ip r(oute)`**	IP routing and state information

For more information about `iproute2`, see the following web pages:

- The Linux Foundation wiki
- IPROUTE2 Utility Suite How to

For more information about the deprecated `net-tools` package, see this web page.

15

Miscellaneous recipes and helpful information

This chapter contains miscellaneous recipes and other helpful information. These topics facilitate administration, save time, increase functionality, or add capabilities to your systems.

This chapter includes the following topics:

15.1 Installing a package from the IBM VM Download Library

The IBM VM Download Library is a clearinghouse or repository for tools, documentation, and other interesting VM topics. The repository is served by a z/VM web server that runs on Conversational Monitor System (CMS).

The IBM z/VM platform development team in Endicott set up the library so that IBM employees, IBM Business Partners, and customer can submit content and so that anyone can download content. The library contains many items, which you are encouraged to explore. IBM provides the library content on an “as-is” basis. IBM might not review all of the library content so you must use your discretion to determine whether a package is appropriate for your environment. If you want to download a package, this section provides the information about downloading. As you become more experienced with the z/VM platform, you might create tools and utilities yourself, which you are encouraged to share.

> **Note:** Before you download anything from the VM Download Library, you must read the license agreement for downloads to ensure that you can comply with the terms.

15.1.1 CMS-based z/VM web browser

To use the CMS web browser, called *Charlotte*, it must be *reblocked* and then, unpacked. Reblocking means reorganizing a file so that it correctly wraps each line.

In section 6.7, “Enabling and configuring RACF” on page 155, the VMARC was placed at `VMPSFS:VMWW2`. Complete the following steps:

1. Log on as either `MAINT` or `MAINT720` on the first node in the cluster.
2. Access `VMPSFS:VMWW2` as `W` read/write and `VMPSFS:MAINT720.UTILS.VMARC` as `V`:

   ```
   ===> access vmpsfs:vmww2. W (forcerw
   ===> access vmpsfs:MAINT720.utils.vmarc V
   ```

3. Run the `VMWW2 VMARC` file through this pipeline:

   ```
   ===> pipe < vmww2 vmarc W | fblock 80 00 | > vmww2 vmarc W f 80
   Ready;
   ```

4. Unpack the contents:

   ```
   ===> VMARC unpack vmww2 W = = W
   ```

5. You can now use these **`grant`** commands to grant access to the browser for all virtual machines:

   ```
   ===> grant auth vmpsfs:vmww2. to public ( read newread
   ===> grant auth * * vmpsfs:vmww2. to public ( read
   ```

   Alternatively, you can substitute individual users in place of **`public`**:

   ```
   ===> grant auth vmpsfs:vmww2. to FMIRANDA ( read newread
   ```

> **Important:** We did not put this data into a Shared File System (SFS) directory under `MAINT` or `MAINT720` because you must not access the web from a master system management virtual machine, such as `MAINT` or `MAINT720`. Use your own class G user ID.

6. You can now use Charlotte by issuing this command from any virtual machine to which you granted authorization:

   ```
   ===> VMLINK .DIR VMPSFS:VMWW2. < . Z-A * > (INVOKE WW2
   ```

15.1.2 Quick and easy display of DIRMAINT directory records

Download QDIR if you use the IBM Directory Maintenance Facility (DirMaint) as your z/VM user directory administration subsystem. QDIR is a tremendous time saver. It quickly and easily shows you the contents of a directory record without having to ask DirMaint for a file, waits for it to be sent, uses peek to view it or receive it, and so on. An example of how QDIR works is shown in Example 15-1.

Example 15-1 Using QDIR to query a user and their associated profile

```
Ready; T=0.01/0.01 00:43:53
QDIR PWNOVAK

USER PWNOVAK PWNOVAK 32M 2G CG                                     10271434
   INCLUDE ATSDFLT                                                 10271434
   IPL CMS PARM FILEPOOL ATS:                                      10271434
   LINK MAINT 0191 0196 MR                                         09171032
   LINK MAINT710 0191 0197 RR                                      09171031
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW0 LNGAMENG PWC20130531 CRCÃé         10080002

Ready; T=0.01/0.01 00:44:23
QDIR ATSDFLT

PROFILE ATSDFLT                                                    12071552
   ACCOUNT ATSDFLT                                                 12071542
   CONSOLE 0009 3215 T OPMGRM1 OBSERVER                            12071542
   SPOOL 000C 2540 READER *                                        12071542
   SPOOL 000D 2540 PUNCH A                                         12071542
   SPOOL 000E 1403 A                                               12071542
   LINK MAINT 0190 0190 RR                                         12071542
   LINK MAINT 019D 019D RR                                         12071542
   LINK MAINT 019E 019E RR                                         12071542
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20140606 CRC U         10080002

Ready; T=0.01/0.01 00:45:22
```

The code is available at this web page.

15.1.3 Automatic closure of spooled consoles

More modern and easy to use solutions are available for capturing console output from virtual machines than spooling to the reader. One such example is Operations Manager for z/VM, where you issue a command from a NAMES file to view the console of another virtual machine, and can even interact with that console in real time.

If you still use the old method of spooling consoles, it is critically important that you have a method to make sure they do not become infinitely large. One way to manage this issue is by installing the code for CONCLOSE into the VMCRON virtual machine.

For more information about CONCLOSE see this web page.

15.1.4 TOOLSRUN

Whatever you store on minidisks and in SFS (such as tools disks, code disks, configurations, and so on) can be shared across multiple systems by using the TOOLSRUN package.

TOOLSRUN allows coordinated set-updates by multiple people to a single disk. The authority to update is based on the permissions granted to you.

For more information about this EXEC, see this web page.

15.1.5 EDEVICE path management

This description is taken from this IBM Products and Solutions web page.

In managing EDEVs, their FCP CHIPIDs, and the SAN controllers hosting the LUNs, it is sometimes wanted to take a component out of service. For example, to apply service to a node of a V7000 SAN storage controller, it is first necessary to take the node out of the configuration. This amounts to removing all EDEV paths that lead to any of the WWPNs of the node. Once the paths are removed from the EDEVs, the V7000 node can be serviced. After the service is complete, it is desirable to put back the removed paths:

- The EDEVPATH tool lets a class B user control the paths of the system's EDEVs en masse. The following functions are provided:
- All of the paths that share a trait, such as a specific FCP CHPID, can be removed from the EDEVs' path configurations. The paths removed are recorded in a log file.
- All of the paths that are removed by an invocation of the "remove" function can be returned. This process is done by reading the log file and undoing the logged removals.
- A snapshot of the system's EDEVs' path configurations can be collected.

The package consists of an exec and a NAMES file. The exec's command line implements the remove, restore, and snapshot functions. The NAMES file holds lists of WWPNs, each list representing a SAN storage controller node. To remove all paths that lead to a specific node, define the node's WWPN list in the NAMES file and then, use the "remove" function, which targets the nicknamed list of WWPNs.

15.2 Modifying the z/VM LOGON panel

There are four major steps to this process:

1. Set up the DRAWLOGO utility

   This step is a one-time process that is done the first time only.
2. Set up the LOGO CONFIG parameters file

   This step is a one-time process that is done the first time only
3. Creating the LOGO file.
4. Placing the new LOGO file into service.

All of the steps in this section must be performed while logged in as `MAINT720`.

> **Note regarding FILELIST:** When you are working in FILELIST and entering command text by typing over a line, if you make a mistake while you are typing, press **F2** to refresh the window and start over. Do not use backspace or delete.

Setting up the DRAWLOGO utility

DRAWLOGO is shipped as a sample exec. Complete the following steps to copy and rename the two files so that they are functional:

1. In 6.11.2, “Copying the utilities to Shared File System file pools” on page 186, you created a directory that is named `UTILS` in the shared VM Product SFS file pool, VMPSFS. You now create a new SFS directory underneath it named `SYSLOGO`:

   ```
   ===> CREATE DIRECTORY VMPSFS:MAINT720.UTILS.SYSLOGO
   ```

2. Use VMLINK to access the new SFS directory as file mode L:

   ```
   ===> ACCESS VMPSFS:MAINT720.UTILS.SYSLOGO L ( FORCERW )
   ```

3. Use VMLINK to access the MAINT 02C2 disk and display the contents in FILELIST:

   ```
   VMLINK MAINT 02C2 < = = RR > ( FILELIST )
   ```

Setting up the LOGO CONFIG parameters file

The `LOGO CONFIG` file must be backed up, edited, and then, copied from the `PMAINT 0CF0` disk to the `MAINT720` A-DISK. Complete the following steps:

1. Use `VMLINK` to access `PMAINT 0CF0` and display the contents with `FILELIST`:

   ```
   ===> VMLINK PMAINT OCFO < = = MR > (FILELIST)
   ```

   The FILELIST panel displays something similar to what is shown in Example 15-2.

   Example 15-2 Display of the PMAINT 0CF0 disk

   ```
   MAINT720 FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
   Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
       LOGO     CONFIG   Y1 V          69          63          1 2020-06-24 09:14:16
       SYSTEM   CONFIG   Y1 F          80         368          8 2020-10-05 11:36:03
   ```

2. Create a copy of `LOGO CONFIG` from the file by moving your cursor to the beginning of the `LOGO CONFIG` line, entering the following text, and then, pressing **Enter**. You type over part of the existing text, as shown in Example 15-3.

   ```
   COPY / LOGO-1 = = (OLDDATE)
   ```

   Example 15-3 Typing over visible text to copy the file

   ```
   MAINT720 FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
   Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
   COPY / LOGO-1 = = (OLDDATE)     69          63          1 2020-06-24 09:14:16
       SYSTEM   CONFIG   Y1 F          80         368          8 2020-10-05 11:36:03
   ```

3. Press **F2** to refresh your display. It should look similar to Example 15-4.

   Example 15-4 Refreshed display

   ```
   MAINT720 FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=6
   Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
       LOGO     CONFIG   Y1 V          69          63          1 2020-10-08 15:03:14
       SYSTEM   CONFIG   Y1 F          80         368          8 2020-10-05 11:36:03
       LOGO-1   CONFIG   Y1 V          69          63          1 2020-06-24 09:14:16
   ```

4. Edit the original text by moving your cursor to the beginning of the line displaying `LOGO CONFIG` and entering the command XEDIT, or pressing **F11**:

   ```
   COPY / = = = (OLDDATE
   ```

Putting the log on panel in place

Complete the following steps:

1. Query the system to show the list of minidisks that are owned by the z/VM Control Program:

   ```
   ===> CP QUERY CPDISK
   ```

   The output should look similar to that shown in Example 15-5.

Example 15-5 Output from CP QUERY CPDISK

```
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  VMBRES 95B2 CKD          39        158
MNTCF3 MAINT    0CF3  C   R/O  VMBRES 95B2 CKD         160        279
```

2. Issue a release request for the C-DISK:

   ```
   ===> CP CPRELEASE C
      CPRELEASE request for disk C scheduled.
      HCPZAC6730I CPRELEASE request for disk C completed.
      Ready;
   ```

3. Use VMLINK to copy the new logo file from the A-DISK to the MAINT 0CF3 disk:

   ```
   ===> VMLINK MAINT 0CF3 < = = MR > (INVOKE COPYFILE ZVMCLOUD LOGO A = = .FM (OLDDATE))
      DMSVML2060I MAINT 0CF3 linked MR as 0120 file mode Z
      DMSVML2061I MAINT 0CF3 detached
      Ready;
   ```

4. Use VMLINK to bring up a FILELIST panel of the MAINT 0CF3 disk. Example 15-6 shows something similar to what you should see.

   Example 15-6 FILELIST displaying the MAINT 0CF3 disk

   ```
   MAINT720 FILELIST A0  V 169  Trunc=169 Size=15 Line=1 Col=1 Alt=0
   Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
       ZVMCLOUD LOGO     Z1 F         78         22          1 2020-10-08 09:57:15
       CPBASE   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
       CPLOAD   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
       ICKSADSF MODULE   Z2 V      65535         18        268 2020-06-26 08:59:38
       SALIPL   MODULE   Z2 V      65535          6         51 2020-06-24 10:17:44
       DDR      MODULE   Z2 V      65535          9        110 2020-06-24 10:17:41
       ESAIO    MODULE   Z2 V      65535         10        112 2020-06-24 10:17:41
       MINIMUM  LOGO     Z1 F         78         16          1 2001-08-17 15:26:04
       PRINTSEP LOGO     Z1 F         49         16          1 2001-08-17 15:25:49
       DEFAULT  LOGO     Z1 F         78         15          1 2001-08-17 15:24:42
       SNA      LOGO     Z1 F         78         15          1 2001-08-17 15:24:18
       LDEV     LOGO     Z1 F         78         15          1 2001-08-17 15:24:06
       LOCAL    LOGO     Z1 F         78         15          1 2001-08-17 15:23:44
       ONMESS   SAMPLE   Z1 F         78          1          1 2000-10-18 21:47:32
       INPTAREA SAMPLE   Z1 F         78          6          1 1993-04-27 18:27:26
   ```

5. Make sure that you see your new LOGO file, that the format is F to indicate Fixed length, and LRECL is 78.
6. (Optional) Update the ONLINE message for the cluster or system by completing the following steps:

   a. Create a file named ZVMCLOUD ONLINE from the existing ONMESS SAMPLE file by moving your cursor to the beginning of the ONMESS SAMPLE line, entering the following text, and then, pressing **Enter**. You type over part of the existing text, as shown in Example 15-7.

If you make a mistake while you are typing, press **F2** and start over. Do not use backspace or delete.:

COPY / ZVMCLOUD ONLINE = (OLDDATE

Example 15-7 Typing over visible text on the FILELIST panel

```
MAINT720 FILELIST A0  V 169  Trunc=169 Size=16 Line=1 Col=1 Alt=28
Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
    ZVMCLOUD LOGO     Z1 F         78         22          1 2020-10-08 09:57:15
    CPBASE   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
    CPLOAD   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
    ICKSADSF MODULE   Z2 V      65535         18        268 2020-06-26 08:59:38
    SALIPL   MODULE   Z2 V      65535          6         51 2020-06-24 10:17:44
    DDR      MODULE   Z2 V      65535          9        110 2020-06-24 10:17:41
    ESAIO    MODULE   Z2 V      65535         10        112 2020-06-24 10:17:41
    MINIMUM  LOGO     Z1 F         78         16          1 2001-08-17 15:26:04
    PRINTSEP LOGO     Z1 F         49         16          1 2001-08-17 15:25:49
    DEFAULT  LOGO     Z1 F         78         15          1 2001-08-17 15:24:42
    SNA      LOGO     Z1 F         78         15          1 2001-08-17 15:24:18
    LDEV     LOGO     Z1 F         78         15          1 2001-08-17 15:24:06
    LOCAL    LOGO     Z1 F         78         15          1 2001-08-17 15:23:44
COPY / ZVMCLOUD ONLINE = (OLDD     78          1          1 2000-10-18 21:47:32
    INPTAREA SAMPLE   Z1 F         78          6          1 1993-04-27 18:27:26
```

b. After pressing **Enter**, press **F2** to refresh your display. It should look similar to Example 15-8.

Example 15-8 Results of the copy process

```
MAINT720 FILELIST A0  V 169  Trunc=169 Size=17 Line=1 Col=1 Alt=56
Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
    ZVMCLOUD LOGO     Z1 F         78         22          1 2020-10-08 09:57:15
    CPBASE   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
    CPLOAD   MODULE   Z1 V      65535        192       3052 2020-06-26 09:02:09
    ICKSADSF MODULE   Z2 V      65535         18        268 2020-06-26 08:59:38
    SALIPL   MODULE   Z2 V      65535          6         51 2020-06-24 10:17:44
    DDR      MODULE   Z2 V      65535          9        110 2020-06-24 10:17:41
    ESAIO    MODULE   Z2 V      65535         10        112 2020-06-24 10:17:41
    MINIMUM  LOGO     Z1 F         78         16          1 2001-08-17 15:26:04
    PRINTSEP LOGO     Z1 F         49         16          1 2001-08-17 15:25:49
    DEFAULT  LOGO     Z1 F         78         15          1 2001-08-17 15:24:42
    SNA      LOGO     Z1 F         78         15          1 2001-08-17 15:24:18
    LDEV     LOGO     Z1 F         78         15          1 2001-08-17 15:24:06
    LOCAL    LOGO     Z1 F         78         15          1 2001-08-17 15:23:44
    ONMESS   SAMPLE   Z1 F         78          1          1 2000-10-18 21:47:32
    ZVMCLOUD ONLINE   Z1 F         78          1          1 2000-10-18 21:47:32
    INPTAREA SAMPLE   Z1 F         78          6          1 1993-04-27 18:27:26
```

c. Move your cursor to the beginning of the line displaying ZVMCLOUD ONLINE and press **F11** to open the file in XEDIT.

In our example environment, we changed the contents of the file as shown in Example 15-9.

Example 15-9 Contents of ZVMCLOUD ONLINE file

```
ZVMCLOUD ONLINE   Z1  F 78  Trunc=78 Size=1 Line=0 Col=1 Alt=3
```

```
00000 * * * Top of File * * *
00001 z/VM Enterprise Cloud Virtualization Platform  |  AUTHORIZED USERS ONLY
00002 * * * End of File * * *
```

d. Enter `FILE` at the XEDIT command line to save and quit:

```
====> FILE
```

7. Press F3 to exit from FILELIST:

```
DMSVML2061I MAINT 0CF3 detached
Ready;
```

8. Return the MAINT 0CF3 disk to service by running the following command:

```
===> CP CPACCESS MAINT 0CF3 C SR
   CPACCESS request for mode C scheduled.
   Ready;
   HCPZAC6732I CPACCESS request for MAINT's 0CF3 in mode C completed.
```

9. If you do not see the message HCPZAC6732I from the last step, stop here and troubleshoot. Never release both CPDISK A and CPDISK C at the same time for any reason.
10. If you see that your CPACCESS request completed successfully, issue the following commands to synchronize the changes you made to CPDISK C onto CPDISK A:
   - CP CPRELEASE A
   - CPRELEASE request for disk A scheduled.
   - HCPZAC6730I CPRELEASE request for disk A completed.
   - Ready; T=0.01/0.01 13:05:38
   - VMLINK MAINT 0CF1 < = X MR > (NON
   - DMSVML2060I MAINT 0CF1 linked MR as 0120 file mode X
   - Ready; T=0.01/0.01 13:09:42
   - VMLINK MAINT 0CF3 < = Y RR > (NON
   - DMSVML2060I MAINT 0CF3 linked RR as 0121 file mode Y
   - Ready; T=0.01/0.01 13:09:57
   - COPYFILE ZVMCLOUD * Y = = X (OLDDATE)
   - Ready; T=0.01/0.01 13:10:46
   - LISTFILE ZVMCLOUD * * (ISODATE
   - FILENAME FILETYPE FM FORMAT LRECL RECS BLOCKS DATE TIME
   - ZVMCLOUD LOGO A1 F 78 22 1 2020-10-08 09:57:15
   - ZVMCLOUD LOGO X1 F 78 22 1 2020-10-08 09:57:15
   - ZVMCLOUD ONLINE X1 F 78 1 1 2020-10-08 12:53:55
   - ZVMCLOUD LOGO Y1 F 78 22 1 2020-10-08 09:57:15
   - ZVMCLOUD ONLINE Y1 F 78 1 1 2020-10-08 12:53:55
   - Ready; T=0.01/0.01 13:11:53
   - ZVMCLOUD LOGO should have the same values for FORMAT, LRECL, BLOCKS, DATE, and TIME for file modes A, X, and Y.

– ZVMCLOUD ONLINE should have the same values for FORMAT, LRECL, BLOCKS, DATE, and TIME for file modes X and Y.

If everything looks correct, it is time to put the CPDISK A back into service by using the following commands:

– RELEASE X (DETACH)
– RELEASE Y (DETACH)
– CP CPACCESS MAINT 0CF1 A SR
– CPACCESS request for mode A scheduled.
– Ready; T=0.01/0.01 13:17:40
– HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.

Next, we must tell the system to begin using these new files. Use VMLINK to access the PMAINT 0CF0 disk:

```
VMLINK PMAINT 0CF0 < = = MR > ( FILELIST
```

You should see something similar to Example 15-10.

Example 15-10

```
MAINT720 FILELIST A0  V 169  Trunc=169 Size=2 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
    SYSTEM   CONFIG   Z1 F         80        368          8 2020-10-05 11:36:03
    LOGO     CONFIG   Z1 V         69         63          1 2020-06-24 09:14:16
```

11. Because the contents of all the following files are identical, we do not need to make backups of all of them; however, you might want to do so:

```
– MINIMUM LOGO
– DEFAULT  LOGO
– SNA      LOGO
– LDEV     LOGO
– LOCAL    LOGO
```

12. Make a backup copy of the `SYSTEM DTCPARMS` file by moving your cursor to the beginning of the `SYSTEM DTCPARMS U1` line and entering the following text (you type over part of the existing text). If you make a mistake while you are typing, press **F2** and start over; do *not* use Backspace or Delete keys:

```
COPY / = DTCPWRKS = (OLDDATE
```

15.3 Using DirMaint to set special passwords for an ID

If you are using the IBM Directory Maintenance Facility (DirMaint) to manage your z/VM user directory, you want to set a special password status for a USER or IDENTITY.

The following special passwords are available:

- NOLOG

  No log on or authentication is available for this ID. This password is used for any ID that is a model or template, used only to own a system resource, such as a minidisk or SFS directory, or otherwise.

- LBYONLY

  Log on is permitted by way of the surrogate log on or log on by procedure only, which is similar in overall concept to the use of *sudo switch user* in Linux. For example, `sudo su - alternateaccount`

- AUTOONLY

  Log on is permitted only by using the **AUTOLOG** or **XAUTOLOG** commands. Authentication of this ID is not possible.

- NOPASS

  The use of a password is not required. As of this writing, few legitimate uses of this option exist. Do not use NOPASS unless an ID is shipped with this set by IBM during system installation, or you are directed to do so by IBM support.

15.4 Resuming a revoked ID in RACF/VM

Checking the status of an example virtual server ID, LNCG4050 with the command **RAC LISTUSER** is shown in Example 15-11. Notice the line that begins with the text `ATTRIBUTES=REVOKED`. This information verifies that the status of this ID is `REVOKED`.

Example 15-11 RACF/VM LISTUSER output

```
RAC LISTUSER LNCG4050
USER=LNCG4050  NAME=UNKNOWN  OWNER=DIRMAINT  CREATED=20.261
 DEFAULT-GROUP=SYS1     PASSDATE=N/A    PASS-INTERVAL=N/A PHRASEDATE=N/A
→ ATTRIBUTES=REVOKED
 ATTRIBUTES=PROTECTED
 REVOKE DATE=NONE   RESUME DATE=NONE
 LAST-ACCESS=20.280/13:18:55
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
 LOGON ALLOWED (DAYS)          (TIME)
 ---------------------------------------------
 ANYDAY                          ANYTIME
  GROUP=SYS1      AUTH=USE      CONNECT-OWNER=DIRMAINT  CONNECT-DATE=20.261
    CONNECTS=    03  UACC=NONE     LAST-CONNECT=20.280/13:18:55
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
```

To set the ID back to a functional status, or *resume* the ID, the RACF/VM command **RAC ALTUSER** is issued to modify an attribute:

```
===> RAC ALTUSER LNCG4050 RESUME
Ready; T=0.01/0.01 13:49:52
```

Finally, verify that the status of the attribute changed by again issuing the **RACF LISTUSER** command, as shown in Example 15-12. Notice that the line that includes `ATTRIBUTES=REVOKED` was removed.

Example 15-12 RACF LISTUSER showing resumed ID

```
RAC LISTUSER LNCG4050
USER=LNCG4050  NAME=UNKNOWN  OWNER=DIRMAINT  CREATED=20.261
 DEFAULT-GROUP=SYS1     PASSDATE=N/A    PASS-INTERVAL=N/A PHRASEDATE=N/A
 ATTRIBUTES=PROTECTED
 REVOKE DATE=NONE   RESUME DATE=NONE
 LAST-ACCESS=20.316/13:49:52
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
 ---------------------------------------------
 ANYDAY                          ANYTIME
  GROUP=SYS1      AUTH=USE      CONNECT-OWNER=DIRMAINT  CONNECT-DATE=20.261
    CONNECTS=    03  UACC=NONE     LAST-CONNECT=20.280/13:18:55
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
```

15.5 System modifications for wide-screen terminals

As you become more proficient with z/VM, you find that the use of a 3270 emulator that is set to a low number of rows and columns can be frustrating. Most 3270 emulators offer the following default sizes, which match up with what used to be IBM 3270 - 3279 machine types:

- 80x24
- 80x32
- 80x43
- 132x27
- 160x43

You find that a larger size allows you to be much more productive, and can display much more information. It also provides the opportunity to use more advanced tools, such as Fullscreen CMS.

Fullscreen CMS

For more information about Fullscreen CMS, see this web page.

DirMaint modifications

Example 15-13 shows a snippet from a `CONFIGAA DATADVH` file that was used to override DirMaint defaults, which makes the output lines longer to eliminate wrapping. This change overrides the 52 or 73 character line length. DirMaint displays output lines of 127 characters before wrapping them. For a 160-column screen, 127 characters works well.

Example 15-13 Overriding DirMaint line size

```
00053 // By default, DirMaint will dynamically select a message output length
00054 // of either 52 or 73 characters.  User's may select a "language" whose
00055 // messages are formatted for a line length other than the default.
00056 // Note:  The maximum linesize is equal to 222; because the maximum
```

```
// length of the CP command buffer is 240, minus 9 for the userid and
// intervening blank, minus 10 for the "CP MSGNOH" command and another
// blank.  The minimum linesize is 40.
// SAMPL_LINESIZE_140A= 222
// SAMPL_LINESIZE_150A= 222
   AMENG_LINESIZE_140A= 127
   AMENG_LINESIZE_150A= 127
```

The output from a command with longer lines showing all of the wrapping is shown in Example 15-14.

Example 15-14 Typical 73 character line wrapping

```
DIRMAINT QUERY DVHLEVEL
DVHXMT1191I Your QUERY request has been sent for processing to
DVHXMT1191I DIRMAINT at *.
Ready; T=0.01/0.01 09:41:03
 DVHREQ2288I Your QUERY request for PWNOVAK at * has
 DVHREQ2288I been accepted.
 DVHQRY3844I Service machine DIRMAINT at node * is
 DVHQRY3844I currently running:
 DVHQRY3844I IBM Directory Maintenance Facility
 DVHQRY3844I for z/VM (DirMaint)
 DVHQRY3844I 5741-A09 (C) Copyright IBM
 DVHQRY3844I Corporation 1979, 2020.
 DVHQRY3844I Function Level 720 Service Level 0000.
 DVHREQ2289I Your QUERY request for PWNOVAK at * has
 DVHREQ2289I completed; with RC = 0.
```

The output from the same command after having made the change to override the line length is shown in Example 15-15.

Example 15-15 Output with overriding the line length

```
DVHREQ2288I Your CMS request for DIRMAINT at * has been accepted.
DVHCMS3868I FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
DVHCMS3868I LINUX    PROTODIR E2 V         37          4          1 2013-04-25 14:19:45
DVHCMS3868I LINUX    PROTODIR G2 V         37          4          1 2013-04-25 14:19:45
DVHREQ2289I Your CMS request for DIRMAINT at * has completed; with RC = 0.
```

To override the line length on a system that is set up and running, you must determine which DirMaint override files are in place. If you used this book to setup your z/VM system, it is likely that you have only the single configuration override, as described in 10.2.2, “Configuring DirMaint” on page 311.

However, it often is the case that multiple configuration files are shown, the one with the highest number is the one that you modify. Therefore, if the output that is shown in Example 15-16 listed a CONFIGAA and CONFIG99, the values that are set in CONFIG99 override anything that was set in the default configuration file, CONFIG DATADVH, and any lower priority override files, such as CONFIGAA DATADVH.

Example 15-16 Example output from DirMaint to list configuration files

```
DIRMAINT CMS LISTFILE CONFIG* DATADVH * (ISODATE)
DVHXMT1191I Your CMS request has been sent for processing to DIRMAINT at
DVHXMT1191I *.
Ready; T=0.01/0.01 11:38:24
 DVHREQ2288I Your CMS request for MAINT at * has been accepted.
```

```
DVHCMS3868I FILENAME FILETYPE FM FORMAT LRECL       RECS
DVHCMS3868I BLOCKS     DATE     TIME
DVHCMS3868I CONFIG   DATADVH  D2 V         73       1648         27
DVHCMS3868I 2018-06-22 13:37:31
DVHCMS3868I CONFIGAA DATADVH  D2 V         73         67          2
DVHCMS3868I 2020-09-01 12:47:03
DVHREQ2289I Your CMS request for MAINT at * has completed; with RC = 0.
```

The output that is shown in Example 15-16 makes it clear that CONFIGAA is the only configuration override file that is used.

Request DirMaint to send CONFIGAA DATADVH so it can be edited:

```
===> DIRMAINT SEND CONFIGAA DATADVH
   DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT
   DVHXMT1191I at *.
   Ready; T=0.01/0.01 11:52:35
    DVHREQ2288I Your SEND request for MAINT720 at * has been accepted.
→ RDR FILE 0276 SENT FROM DIRMAINT PUN WAS 0342 RECS 0059 CPY  001 A NOHOLD
NOKEEP
    DVHREQ2289I Your SEND request for MAINT at * has completed; with RC = 0.
```

Make note of the line that begins with `RDR FILE`, which indicates the reader file number that was sent. You need this number to retrieve the file from the reader, as shown in Example 15-17. You must replace the example value `0276` with the correct value for your system.

Example 15-17 Retrieval of CONFIGAA from the reader list

```
DIRMAINT SEND dirmaint send configaa datadvh
DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT
DVHXMT1191I at ICPZVM.
Ready; T=0.01/0.01 11:52:35
 DVHREQ2288I Your SEND request for MAINT at * has been accepted.
RDR FILE 0276 SENT FROM DIRMAINT PUN WAS 0342 RECS 0059 CPY  001 A NOHOLD NOKEEP
 DVHREQ2289I Your SEND request for MAINT at * has completed; with RC = 0.
```

In the situation where an old copy of CONFIGAA DATADVH was left behind on the MAINT720 A-Disk, you see output that is similar to the output that is shown in Example 15-18. In general, it is considered bad practice to leave old copies of any directory configuration files lingering on the MAINT720 A-Disk.

Example 15-18 CONFIGAA exists on the MAINT720 A-Disk

```
RECEIVE 0276 = = A
DMSDDL024E File CONFIGAA DATADVH A2 already exists; specify REPLACE option
DMSDDL1124W Spool file 0276 has been left in your reader
DMSDDL1124W because one or more files were not received
Ready(00028); T=0.01/0.01 12:01:53

RENAME CONFIGAA DATADVH A CONFAA-1 = =
Ready; T=0.01/0.01 12:02:10
```

This example shows how to rename the old copy of the file to CONFAA-1 DATADVH, then receive the current version. It is important that you review CONFAA-1 to see if it contains work that is still in progress by someone, or other changes that were not sent to the DirMaint service machine. If necessary, merge any required values from CONFAA-1 into CONFIGAA before you issue the command to send it back for processing.

15.6 Manually formatting DASD for use

Often, you must add a DASD to the system for extra paging, spooling, or other reasons. During the initial setup of z/VM, we mentioned that certain types of DASD require formatting parameters that determine the ownership of the volume.

The following example covers the formatting of three extra DASDs:

```
===> query 30D0-30D2
DASD 30D0 VM30D0  , DASD 30D1 VM30D1  , DASD 30D2 VM30D2
===> attach 30D0-30D2 to *
30D0-30D2 ATTACHED TO MAINT720
```

Allocate `30D0` to `ITSOZVM2` as a SPOOL volume:

```
===> cpfmtxa 30D0
ENTER FORMAT, ALLOCATE, LABEL, OWNER OR QUIT:
format
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 30D0 OR QUIT:
0-END
ENTER THE VOLUME LABEL FOR DISK 30D0:
VP30D0
CPFMTXA:
FORMAT WILL ERASE CYLINDERS 000000000-000003338 ON DISK 30D0
DO YOU WANT TO CONTINUE? (YES | NO)
yes
HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0             TIME: 19:37:30
05/21/15     PAGE   1

ENTER INPUT COMMAND:
  CPVOL FMT MODE(ESA) UNIT(30D0) VOLID(VP30D0) NOVFY NFILL -
ENTER INPUT COMMAND:
 RANGE(0,3338)
ICK00700I DEVICE INFORMATION FOR 30D0 IS CURRENTLY AS FOLLOWS:
          PHYSICAL DEVICE = 3390
          STORAGE CONTROLLER = 3990
          STORAGE CONTROL DESCRIPTOR = E9
          DEVICE DESCRIPTOR = 0A
          ADDITIONAL DEVICE INFORMATION = 4A001F3C
          TRKS/CYL = 15, # PRIMARY CYLS = 3339
ICK04000I DEVICE IS IN SIMPLEX STATE
ICK00091I 30D0 NED=002107.900.IBM.75.0000000AKAZ1
ICK091I   30D0 NED=002107.900.IBM.75.0000000AKAZ1
ICK03020I CPVOL WILL PROCESS 30D0 FOR VM/ESA MODE
```

```
ICK03090I VOLUME SERIAL = VS30D0
ICK03022I FORMATTING THE DEVICE WITHOUT FILLER RECORDS
ICK03011I CYLINDER RANGE TO BE FORMATTED IS 0 - 3338
ICK003D REPLY U TO ALTER VOLUME 30D0 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 30D0 FOLLOWS:

FORMATTING OF CYLINDER 0 STARTED AT: 19:37:30
         FORMATTING OF CYLINDER 100 ENDED AT: 19:37:30
         FORMATTING OF CYLINDER 200 ENDED AT: 19:37:31
         FORMATTING OF CYLINDER 300 ENDED AT: 19:37:32
         FORMATTING OF CYLINDER 400 ENDED AT: 19:37:32
         FORMATTING OF CYLINDER 500 ENDED AT: 19:37:33
...

          VOLUME SERIAL NUMBER IS NOW = VP30D0

          CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
          TYPE     START      END        TOTAL
          ----     -----      ---        -----
PERM     0          3338       3339

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
          19:38:49    05/21/15

ENTER INPUT COMMAND:
 END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE CYLINDERS
.................
SPOL 0 END
END
HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0 ...

ENTER INPUT COMMAND:
  CPVOL ALLOC MODE(ESA) UNIT(30D0) VFY(VP30D0) -
ENTER INPUT COMMAND:
        TYPE((SPOL,0,3338))
ICK00700I DEVICE INFORMATION FOR 30D0 IS CURRENTLY AS FOLLOWS:
          PHYSICAL DEVICE = 3390
          STORAGE CONTROLLER = 3990
          STORAGE CONTROL DESCRIPTOR = E9
          DEVICE DESCRIPTOR = 0A
          ADDITIONAL DEVICE INFORMATION = 4A001F3C
          TRKS/CYL = 15, # PRIMARY CYLS = 3339
ICK04000I DEVICE IS IN SIMPLEX STATE
ICK00091I 30D0 NED=002107.900.IBM.75.0000000AKAZ1
ICK091I   30D0 NED=002107.900.IBM.75.0000000AKAZ1
```

```
ICK03020I CPVOL WILL PROCESS 30D0 FOR VM/ESA MODE
ICK03090I VOLUME SERIAL = VP30D0
ICK03024I DEVICE IS CURRENTLY FORMATTED WITHOUT FILLER RECORDS
ICK003D REPLY U TO ALTER VOLUME 30D0 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 30D0 FOLLOWS:

          CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
          TYPE     START      END        TOTAL
          ----     -----      ---        -----
          SPOL     0          3338       3339

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
          19:39:55    05/21/15

ENTER INPUT COMMAND:
 END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

===> cpfmtxa 30D0
ENTER FORMAT, ALLOCATE, LABEL, OWNER OR QUIT:
owner
```

15.7 Running Linux under z/VM with restricted permissions

For more information about this concept, see *Running Linux Guest in less than CP Privilege Class G*, REDP-3870. This paper takes a defensive approach to the planning and deployment of a z/VM system that is hardened to enhance security boundaries. Access to any z/VM commands or facilities is restricted to only those that are essential to run Linux.

As the paper mentions, this scenario might not be necessary or wanted for all Linux virtual servers:

> *The guidelines in this paper allow you to run Linux virtual machines in a safer way. This does not necessarily mean that you are required to implement all recommendations, or that you must run all Linux virtual machines that way. If you are confident enough that for some Linux servers the risks are less, then you may decide to lower the fences for those Linux virtual machines. As always, you must make the tradeoff between security, flexibility, and ease of use.*

15.8 Mitigating SSH client timeout disconnects

Often times when a firewall exists between your workstation and a Linux server, sessions that are seen by the firewall as idle can be forcefully severed by the firewall, or even by Linux in some scenarios. This situation can be incredibly frustrating to someone who is trying to administer the system and is forced to focus their attention elsewhere for some amount of time. It is possible to modify the default keep alive timeout value for your local SSH client in an attempt to help prevent forced disconnections, such as the one shown in Example 15-19.

Example 15-19 A forced disconnect of an SSH session due to timeout

```
· ATS Intranet GZ-ATS-ENDZLAB endicott1415 | Thu Nov 05 22:39 UTC
· pwnovak histref=85 pwd=/srv/ftp/ibm/zvm/720/
······> ls -al
total 19
drwxrwx---  9 ftpadmin ftpadmin 4096 2020-09-18 17:28 .
drwxrwx--- 27 ftpadmin ftpadmin 4096 2020-09-18 17:22 ..
drwxrwx---  3 ftpadmin ftpadmin 4096 2020-09-18 17:30 eckd
dr-xr-x---  1 ftpadmin ftpadmin  528 2020-08-11 13:27 isodoc
dr-xr-x---  1 ftpadmin ftpadmin  112 2020-08-11 09:18 isoeckd-1
dr-xr-x---  1 ftpadmin ftpadmin  112 2020-08-11 09:41 isoeckd-2
dr-xr-x---  1 ftpadmin ftpadmin  112 2020-08-11 09:56 isoscsi-1
dr-xr-x---  1 ftpadmin ftpadmin  112 2020-08-11 10:22 isoscsi-2
drwxrwx---  3 ftpadmin ftpadmin 4096 2020-09-18 17:33 scsi

· ATS Intranet GZ-ATS-ENDZLAB endicott1415 | Thu Nov 05 22:40 UTC
· pwnovak histref=86 pwd=/srv/ftp/ibm/zvm/720/
······>
client_loop: send disconnect: Broken pipe
```

On workstations that are Linux or UNIX based (or derivatives thereof, including MacOS X), the potential solution uses the SSH configuration file. This file is in the home directory for your login shell account under the `.ssh` subdirectory.

If you are unsure what the home directory for your account is, run the **env** command to display all of the environment variables set for your login shell, as shown in the following two examples.

In both examples, the login shell in use is `bash`, which is the Bourne Again (modernized) derivative of the Bourne shell, `sh`.

Example 15-20 shows output from a MacOS system.

Example 15-20 Finding the $HOME directory

```
===> env
TERM_PROGRAM=Apple_Terminal
MACOS_VERSION=10.15.7+19H2
TERM=xterm-256color
SHELL=/bin/bash
HISTSIZE=999990000
TMPDIR=/var/folders/3y/...
TERM_PROGRAM_VERSION=433
TERM_SESSION_ID=369AF504-...
USER=pwnovak
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.Eev2DQxR1N/Listeners
```

```
PATH=/opt/local/bin:/opt/local/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/
usr/local/gsa/bin:/Applications/VMwareFusion.app/Contents/Public:/opt/X11/bin:/Lib
rary/Apple/usr/bin:/Users/pwnovak/bin:/usr/local/gsa
LaunchInstanceID=A116603C-...
PWD=/Users/pwnovak/.ssh
DBUS_LAUNCHD_SESSION_BUS_SOCKET=/private/tmp/com.apple.launchd. ...
LANG=en_US.UTF-8
XPC_FLAGS=0x0
HISTIGNORE=&:ls:ls -al:lsa:cd ~:cd ..:[bf]g:exit:h:history:history | grep[.*]
HISTCONTROL=ignoreboth:erasedups
XPC_SERVICE_NAME=0
HOME=/Users/pwnovak
SHLVL=1
LOGNAME=pwnovak
PKG_CONFIG_PATH=/usr/local/opt/icu4c/lib/pkgconfig
DISPLAY=/private/tmp/com.apple.launchd.FWjZwL5eYF/org.macosforge.xquartz:0
SECURITYSESSIONID=...
_=/usr/bin/env
OLDPWD=/Users/pwnovak
```

You can run the following command at a command-line prompt to see whether your ID and HOME directory are set up in your environment variables:

```
echo “My account is ‘$ID’ and the home directory for it is ‘$HOME’...”
```

If you discover that the subdirectory `~/.ssh` (also referenced via `$HOME/.ssh`) does not exist, create it as shown in the following example:

```
===> mkdir -p $HOME/.ssh
```

15.9 Sharing IBM WebSphere Application Server binaries

It is suggested that anyone who is thinking about running WebSphere Application Server Traditional Profile (Network Deployment and stand-alone alike) to explore this topic further.

The IBM Washington Systems Center published the WSC Whitepaper (document ID `WP102730`), which is available at this IBM Support web page, that provides more information about this issue, which includes the following abstract text:

> *This document describes a process that enables you to share one installation of WebSphere Application Server among many Linux guests running under z/VM. This is an update to the previous versions of this document, which covered WebSphere v8.0.x. As a platform for Linux, one of the strengths of IBM Z is the ability to centrally manage many Linux images. But as the number of Linux images grows, some of the typical problems of large server farms emerge. Software installed on the Linux images must be serviced or updated, and there is no way to do this other than servicing each image as if it were a stand-alone server. IBM Z has the ability to share file systems as VM minidisks so there ought to be a way to install WebSphere once and use that installation for many other Linux images. This was previously impractical, because the installed WebSphere program files (which are read/only) and the user data files (which are read/write) were not identified and were intermixed in the same directories. WebSphere Version 6 introduced the concept of "Profiles", which are created by the manage profiles command or graphically with the pmt.sh tool. Profiles must be created for all installations of WebSphere V6 and above, and contain all the user-writable files (logs, installed applications, etc.) for WebSphere. All the profiles on a single server share the same WebSphere binaries from the WebSphere installation directory, so service can be applied to WebSphere and apply to all the profiles. So, with WebSphere V8.5.5.xthe WebSphere binaries are separate from the user data files in the same way as V6, V7 and V8. We can use this separation to allow profiles to share the WebSphere binaries not only on the same server, but across many servers.The methods and procedures for sharing the WebSphere binaries in V8.5.5.xdiffer little from V8. See the following section on what is different for those administrators that are already familiar with the sharing concepts in V8 of this paper.*

Although this document covers a specific version of WebSphere Application Server, the methodologies that are described are substantially similar for newer versions of WebSphere Application Server.

Part 4

Appendixes

This part consists of the following appendixes:

A

Configuring a workstation to deploy and administer z/VM

This appendix addresses configuring a workstation that is running Linux, Apple MacOS, or Microsoft Windows to access z/VM logical partitions and Linux virtual servers. It includes the following topics:

Basic requirements

The following types of clients are required for you to successfully access z/VM and Linux from your workstation to complete the tasks that are described in this book:

- A 3270 terminal emulator with TLS support
- A TTY terminal emulator with SSH support
- A remote graphical display client with support for the Virtual Network Computing (VNC) protocol

Many options are available to provide these three clients. The operating system that is running on your workstation determines which options are available to you. Several programs or products to consider are described for each of the these requirements.

> **Note:** The programs or products that are included in this appendix are not endorsed, nor is their fitness for intended purpose warranted in any way by the authors or publishers of this book.

3270 terminal emulators

The 3270 terminal emulators have been around for some time now. As you consider the options that are available to you for this requirement, be aware that the terminology that is used to refer to these options feature several variations. However, do not allow this issue to confuse you. In general, all of the following terms refer to a 3270 terminal emulator:

- TN3270 client
- Host access client
- Mainframe terminal emulator
- IBM host access client
- Host emulator
- zSeries terminal client

Linux

We recommend the use of x3270 or c3270 to provide this function. Both of these programs, and several other related programs that are part of the overall x3270 suite, are available in the official repositories for Alpine, CentOS, ClefOS, Debian, OpenSUSE, RHEL, SLES, and Ubuntu.

MacOS

We recommend the use of x3270 or c3270 to provide this function. Both of these programs, and several other related programs that are part of the overall x3270 suite, are available by way of MacPorts and Homebrew. If you cannot use MacPorts or Homebrew, the installation code can be found at the x3270 web page.

Windows

Although many options are available for Windows, many users prefer to purchase and use the IBM Personal Communications 3270 emulator. If you have an approved 3270 emulator for your organization, that should be your choice. If you do not, we recommend the Windows version of x3270.

TTY clients

From a Linux workstation, you need the following programs, tools, or utilities:

- A Secure Shell (SSH) client: Any Linux terminal client can perform this function.
- A Virtual Network Computing (VNC) client: `remmina` or `vncviewer` are both equally recommended.
- A 3270 emulator: `x3270` or `c3270` are recommended.

From a Windows workstation, you need the following programs, tools, or utilities:

- A Secure Shell (SSH) client: PuTTY is recommended.
- A Virtual Network Computing (VNC) client: RealVNC is recommended.
- A 3270 emulator: Many choices are available.

PuTTY: A no-charge SSH client for Microsoft Windows

Throughout this book, SSH is used to log in to Linux systems. It is easy to use and cryptographically secure. If you use a Windows desktop, you need a good SSH client. PuTTY is perhaps the most commonly used. You can download PuTTY from this web page.

At this web page, click the **putty.exe** link for your architecture. Save the file in a directory path, such as `C:\WINNT`. PuTTY is a stand-alone executable file. (No installation is needed other than copying the file.) You might also want to create a shortcut on your desktop or taskbar.

Complete the following steps:

1. Open PuTTY, and the configuration window that is shown in Figure A-1 on page 464 opens. If you spend a few minutes configuring PuTTY, it might save time. The examples that are shown use PuTTY Release 0.60.
2. In the PuTTY Configuration window, in the left Category panel, click **Session**.
3. Under the Connection type heading on the right side, click **SSH** (see Figure A-1 on page 464) to use the SSH protocol.

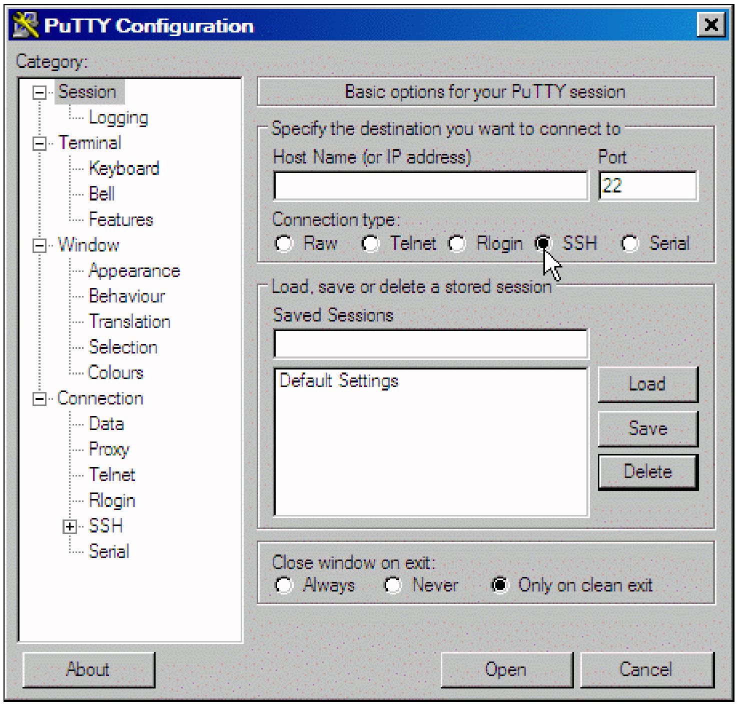


Figure A-1 PuTTY Configuration window

4. Click **Logging** in the left panel, as shown in Figure A-2:
   a. Click **Printable output** in the Session logging radio group to go back and check the output of specific commands.
   b. Set the Log file name to `&H&M&D&T.log` so that a time stamp is in the file name (see Figure A-2).

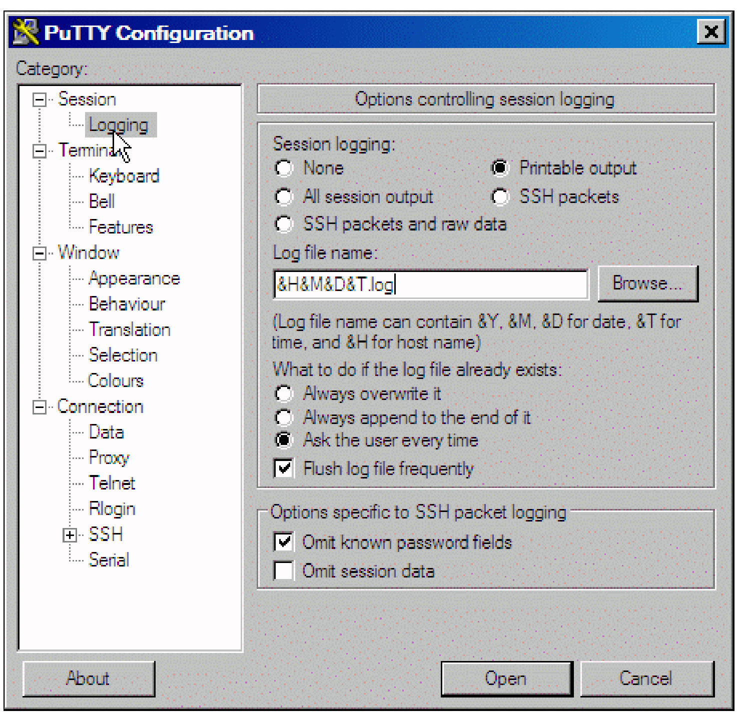


Figure A-2 Setting logging

5. In the left panel, click **SSH** near the bottom, as shown in Figure A-3.

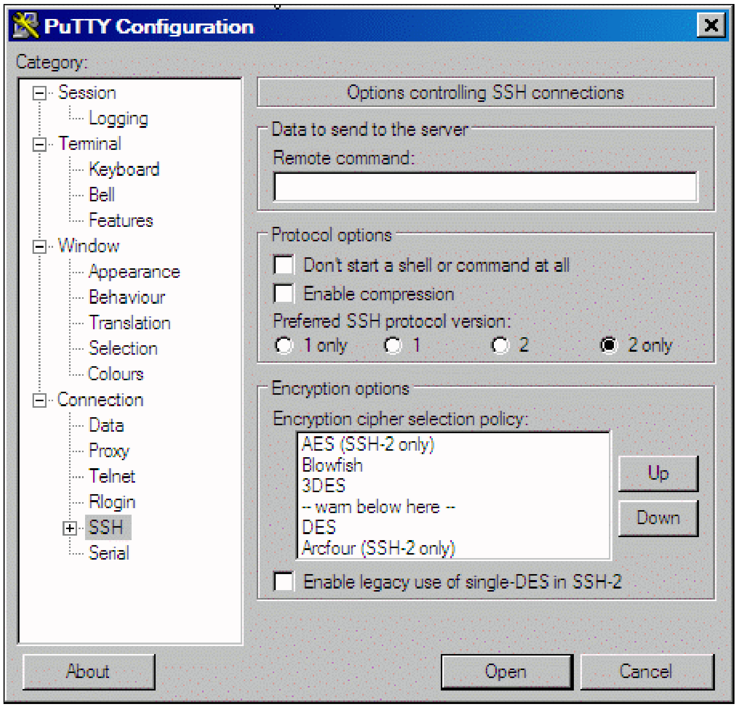


Figure A-3 Setting SSH Protocol 2

6. On the right side, under Preferred SSH protocol version, click **2 only**.

7. In the left Category panel, click **Terminal**, as shown in Figure A-4.

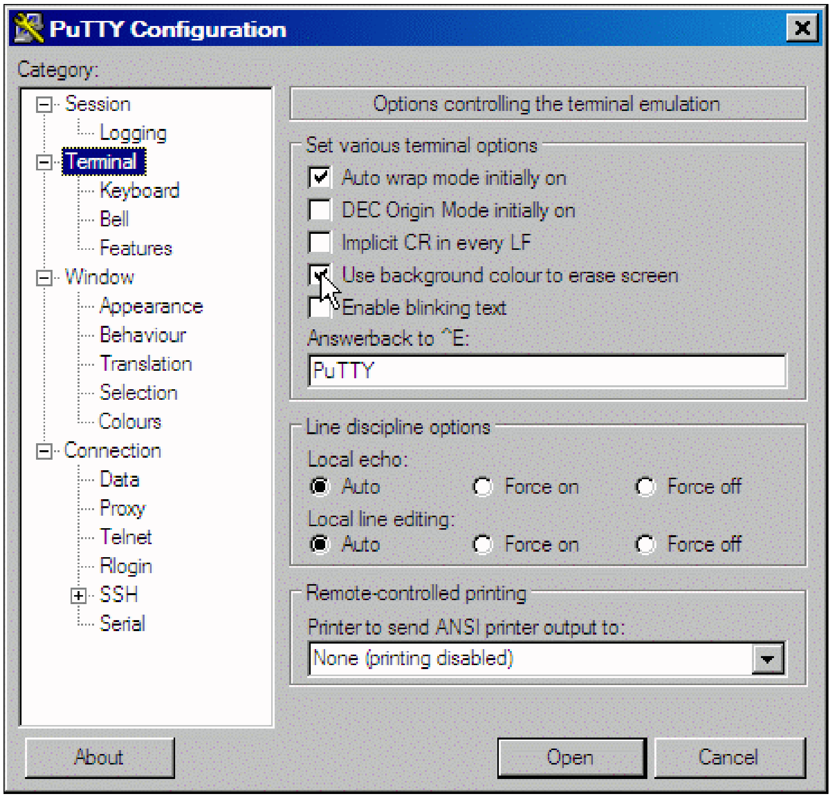


Figure A-4 Customizing PuTTY SSH settings

8. Select **Use background colour to erase screen**, which results in a better job of painting the window for applications that use block graphics.

9. Click **Window** in the left pane, as shown in Figure A-5.

   You can choose a larger window size and more lines of scrollback. In this example, 50 rows, 100 columns, and 1000 lines of scrollback are set.

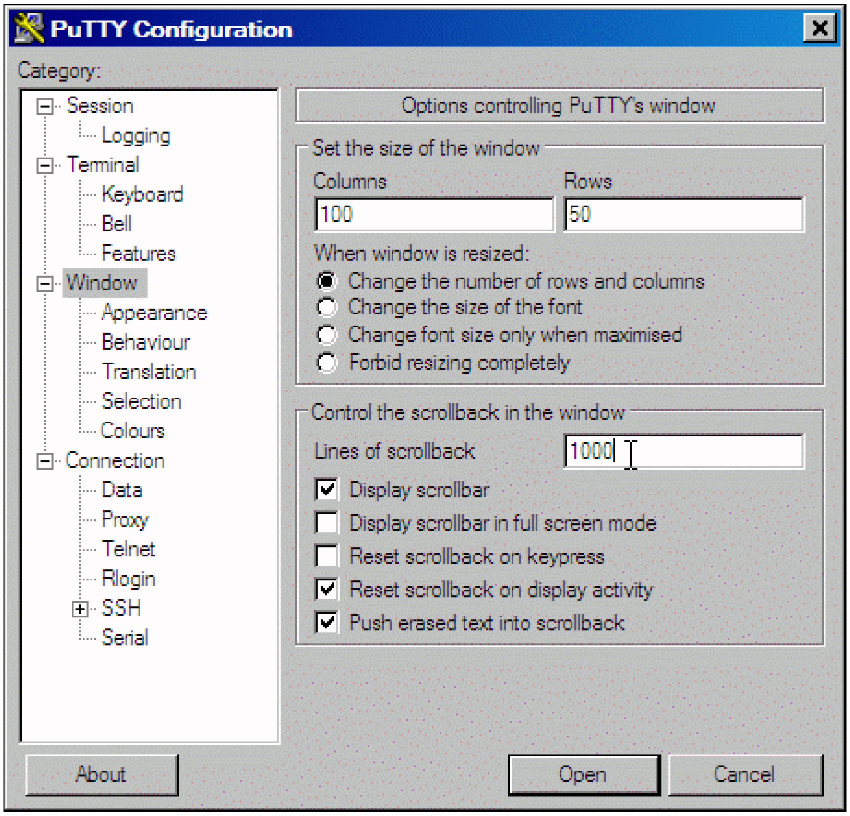


Figure A-5 Setting window and scrollback size

10. Click **Session** in the left pane, as shown in Figure A-6.

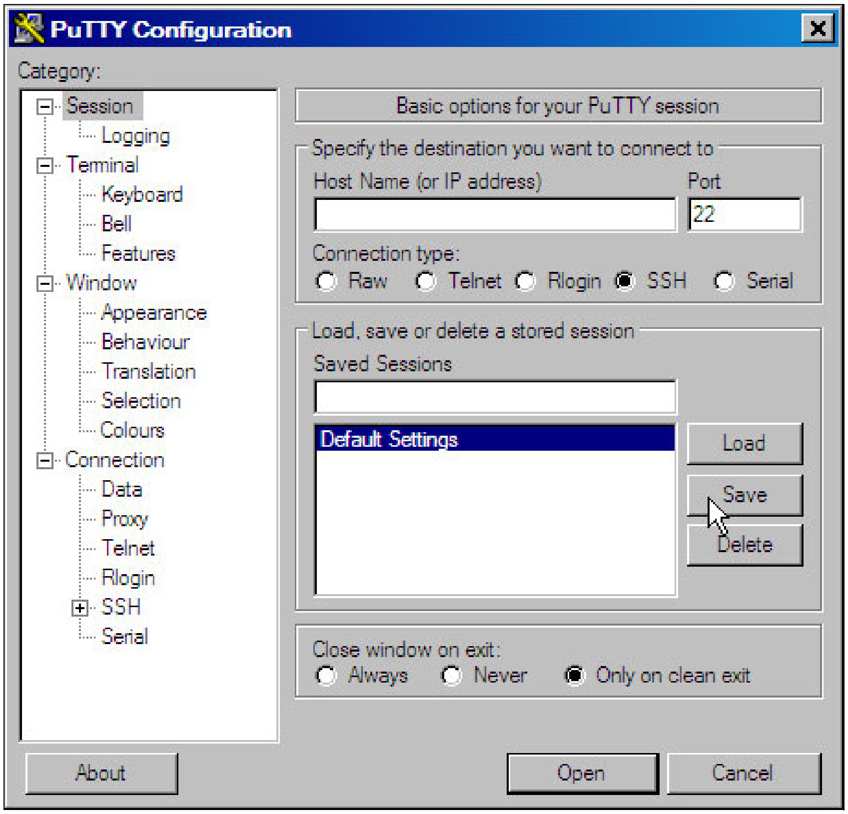


Figure A-6 Saving the new default settings

11. Click **Default Settings** in the Saved Sessions pane. Then, click **Save**. All future sessions that you define inherit the preferences that you set.

Save sessions

In the example that is shown in Figure A-7, a session for `LINUX00` is saved.

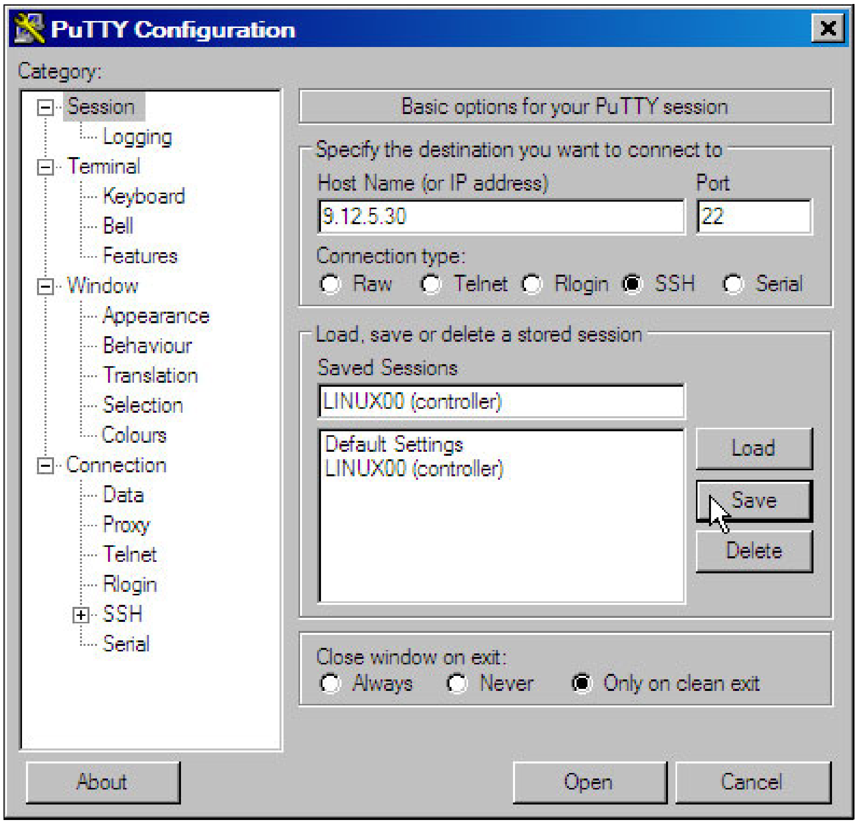


Figure A-7 Customizing PuTTY window settings

To save a session for each virtual machine, complete the following steps:

1. In the Host Name (or IP address) field, enter the TCP/IP address or Domain Name System (DNS) name.
2. Under the Saved Sessions text area, choose a name that you can remember. In this example, the name `LINUX00 (controller)` is used.
3. Click **Save**, and you see the name added to the Saved Sessions list.

   Whenever you start PuTTY, double-click any saved session name, and an SSH session to the Linux system that you want is started.

The use of a Linux workstation instead is the preferred method.

SSH client on macOS and Linux

Linux and macOS provide command line interfaces and the `ssh` client program.

The PuTTY program from Windows, being Open Source, was built on Linux and macOS. Although the author of PuTTY does not build macOS or Linux binaries, they are obtainable, or you can build PuTTY from source.

On macOS, the MacPorts system provides a PuTTY build. The use of PuTTY on a non-Windows system might be convenient if you must use Windows and other systems and want to have a consistent interface across all.

Setting up a VNC client

A VNC client allows access to a graphical user interface environment with Linux on IBM Z.

VNC on Microsoft Windows

If you use a Windows desktop, the VNC client from RealVNC is a popular choice. You can purchase a full function RealVNC client, or a version is available at no charge. For more information, see the RealVNC website.

Complete the following steps:

1. At the RealVNC Download web page, click **Download**. Complete the web form and download the executable file. After you download it, run it, and an installation program starts. At the time of writing of this book, RealVNC 4.1.2 was the current version.
2. Accept all defaults; however, you likely do not need a VNC server on your desktop. Therefore, you can clear **VNC Server** from the Select Components panel, as shown in Figure A-8.

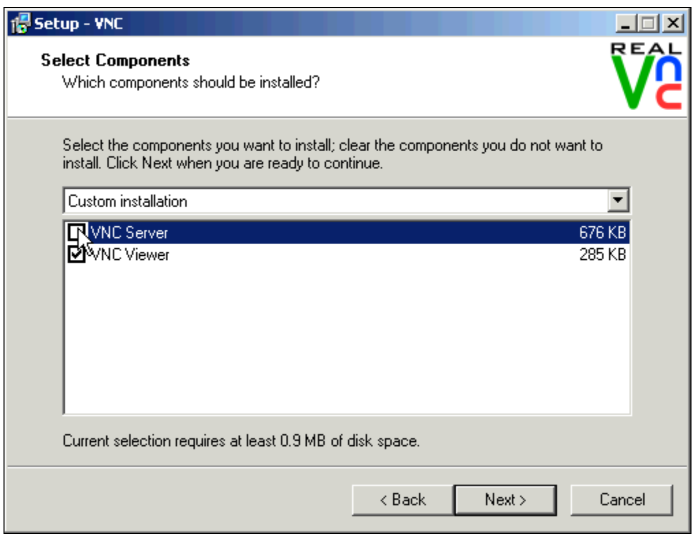


Figure A-8 RealVNC Select Components panel

3. Complete the panels and the installation process goes quickly.

> **Important:** Although no specific download site exists for the RealVNC viewer for Microsoft Windows 7 or 8, instructions for both are available at this web page.
>
> The tool TightVNC might be an option for the Windows operating systems. For more information, see this web page.
>
> TightVNC 2.0.4 supports all client and server versions of Microsoft Windows starting at Windows 2000, up to Windows 7.

Apple macOS

Apple macOS includes a VNC viewer, which is called *Screen Sharing*. It can be used to connect to VNC servers and is run by locating the Screen Sharing icon by using Launchpad or other technique.

You can also use Finder to start a VNC session by using Screen Sharing, by choosing **Go** → **Connect to Server** or pressing **Command-K**. In the window that opens, enter a VNC URL that uses the following format:

```
vnc://<ipaddress_or_hostname>:<port_number>
```

Some alternative VNC viewers also are available. For example, RealVNC provides its VNC client for macOS.

Linux

Linux distributions provide several ways to obtain a VNC viewer. For example, on Red Hat Enterprise Linux, the `tigervnc` package provides the popular TigerVNC client. For the GNOME desktop environment, the Vinagre and Remmina programs also are available.

IBM 3270 emulators

To access a log-on session with z/VM, an emulator for the IBM 3270 terminal is needed. Many alternatives are available, depending on operating system and platform.

Microsoft Windows

Many commercial products are available, including the following common products:

- Micro Focus Attachmate Extra!
- OpenText (formerly Hummingbird) HostExplorer
- IBM Personal Communications
- Rocket BlueZone/Passport PC to Host
- Quick3270
- wc3270 (Windows version of c3270, free/libre Open Source software)

Apple macOS

The following commercial and free options are available:

- EmTec ZOC
- Mocha TN3270 (available via macOS App Store, free trial version also available)
- x3270 suite (x3270, c3270)

x3270 is available by building the package from source or a software packaging system, such as MacPorts or Homebrew. You must ensure that XQuartz is installed so that x3270 can run because it is an X11 program.

> **Note:** The x3270 brew in Homebrew no longer provides `x3270` by default. The Homebrew project removed X11 support from packages in `homebrew-core` because of what was considered to be poor X11 support on macOS. The `c3270` program is provided by the x3270 brew; c3270 provides the functions of x3270 in a terminal window (without requiring X11).
>
> To get the `x3270` program on macOS by using Homebrew, use a command-line option to build the brew from source the includes the **`--with-x11`** option.

Another option on macOS before Catalina (10.15) is the tn3270 X program from Brown University. This native Mac OSX program provides 3270 emulation, but it is a 32-bit application. Apple removed support for 32-bit applications in macOS Catalina, and the maintainers of the program said that they do not have resources to port the program to 64-bit.

Linux

The x3270 suite is the main option for 3270 emulation on Linux. Most distributions provide an x3270 package; for example, the package `x3270-x11` provides the `x3270` program and `x3270-text` provides `c3270` on Red Hat and Fedora.

Mobile applications

Mobile device users can also get 3270 emulation programs. Various options are available, such as Mocha TN3270 and Glink 3270 on iOS, iPadOS, and Android.

Web-to-host platforms

Several systems can provide 3270 access by using a web browser. These systems remove the requirement to have a program installed on a workstation. They also permit access by using Java applets or HTML5-style rendering in a browser. Most of these systems also support the transformation of 3270-style data into more “web-native” formats. These products and systems include the following examples:

- Rocket BlueZone/Passport Web to Host
- h3270 (Open Source)

Tips for choosing and using a 3270 emulator

It is beyond the scope of this book to explain the details of configuring all of the various emulators. However, it is recommended that you investigate the following settings for your emulator:

- Support for encryption.

  Ensure that your emulator can establish a secure connection by using Transport Layer Security (TLS).

- Set the Enter and Clear function keys to be where you expect them.

  On specific emulators, the default Enter key action is set to the right Ctrl key on modern keyboards. Likewise, the Clear key action is sometimes set to the Esc key in the upper-left corner of modern keyboards or the Pause key in the upper right.

- Set a larger window.

  Often, the default number of lines in an emulator session is 24. Productivity likely increases by using a 43-line window (or more) if the lines can easily fit in a window that is based on your desktop display size and resolution.
- Set up the session to automatically reconnect after logging off.

  Opening a new log on window automatically immediately after you log off can also save time. This approach is often not the default behavior.
- Save your connection sessions.

  Rather than continually entering the IP address or DNS name of the z/VM system to which you want to connect, spend a few minutes defining and saving a session for each system to which you can connect, as was described for PuTTY. Then, you often can double-click the saved connection to quickly access a new 3270 session.

Customizing your 3270 emulator on the front end can save time later.

B

Reference, cheat sheets, blank worksheets, and education

This appendix provides extra materials that are included for your reference. These materials can be printed or downloaded, as described. In addition, links to where you can obtain self-learning educational information are included.

This appendix includes the following topics:

- "Related books and publications" on page 476
- "Online resources" on page 477
- "Important z/VM files" on page 478
- "Cheat sheets" on page 479
- "Blank planning worksheet" on page 482

Related books and publications

The following publications can be used as information sources:

- z/VM installation guide

  You want to have a copy of *z/VM 7.2.0 Installation Guide*, GC24-6292, to use as reference.
- Documentation for Linux on IBM Z and LinuxONE is available at this web page.
- z/VM Internet Library

  Online documentation is at this web page.

  The following useful publications are available at this web page:

  – *z/VM CP Messages and Codes*
  – *z/VM TCP/IP Messages and Codes*
  – *z/VM CP Commands and Utilities Reference*
  – *z/VM CP Planning and Administration*
  – *z/VM Getting Started with Linux on System z*
  – *z/VM TCP/IP Planning and Customization*
  – *z/VM Performance Toolkit Guide*, SC24-6156
  – *z/VM Performance Toolkit Reference*, SC24-6157
- z/VM configuration and performance information from Dr. Brian Wade:

  – CPU Utilization in an SMT World
  – Topics in LPAR Performance
- *Linux on IBM Z and LinuxONE Troubleshooting Guide*, SC34-2612
- *z/Architecture Principles of Operation*, SA22-7832

  The authoritative source of detailed information for the how and why IBM Z and LinuxONE systems work; often referred to as *The Z POp*.
- IBM DeveloperWorks guides

  – *How to Improve Performance with PAV (Kernel 2.6.35)*, SC33-8414

  – *How to use FC-attached SCSI devices with Linux on z Systems (Kernel 4.0)*, SC33-8413

  – *How to Set up a Terminal Server Environment*, SC34-2596

  – *Linux Channel Bonding Best Practices and Recommendations*
- Linux390 reference documentation on IBM DeveloperWorks

  The following documents are available at this web page:

  – *Kernel Messages (Kernel 4.19)*, SC34-2599

  – *Pervasive Encryption for Data Volumes*, SC34-2782

  – *Getting Started with Pervasive Disk Encryption*, SC34-2783

  – *libica Programmer's Reference*, SC34-2602

  – *Secure Key Solution with the Common Cryptographic Architecture Application Programmer's Guide*, SC33-8294

  – *Exploiting Enterprise PKCS #11 using openCryptoki*, SC34-2713
- Linux distribution-specific materials: SUSE Linux Enterprise Server 12 on IBM z Systems

- IBM Redbooks publications

  The following IBM Redbooks publications are available at this web page:

  – *IBM z Systems Connectivity Handbook*, SG24-5444
  – *Deploying a Cloud on IBM System z*, REDP-4711
  – *Installing Oracle 11gR2 RAC on Linux on System z*, REDP-4788
  – *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
  – *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266
  – *Security for Linux on System z*, SG24-7728
  – *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995
  – *Set up Linux on IBM System z for Production*, SG24-8137
  – *Practical Migration from x86 to Linux on IBM System z*, SG24-8217
  – *End-to-End High Availability Solution for System z from a Linux Perspective*, SG24-8233
  – *Security for Linux on System z: Securing Your Network*, TIPS0981
  – *Linux on System z: An Ideal Platform to Migrate Your IT Workload*, TIPS1166
  – *Linux on IBM eServer™ zSeries and S/390®: Performance Toolkit for VM*, SG24-6059
  – *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864

Online resources

The following websites and URLs are also relevant as further information sources:

- The z/VM customer presentation library page:

  `http://www.ibm.com/vm/library/presentations/`

- The Linux for z Systems and S/390 portal:

  `http://linuxvm.org`

- The `IBMVM` list server:

  `http://listserv.uark.edu/archives/ibmvm.html`

- The `linux-390` list server:

  `http://www2.marist.edu/htbin/wlvindex?linux-390`

- Red Hat Enterprise Linux Server no-charge evaluation download for IBM z Systems:

  `http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux`

- SUSE Linux Enterprise Server no-charge evaluation download for IBM z Systems:

  `https://www.suse.com/products/server/download/systemz.html`

- z/VM publications:

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

- z/VM performance tips:

  `http://www.vm.ibm.com/perf/tips/`

- z/VM VDISK for Linux swap performance tips:

  `http://www.ibm.com/vm/perf/tips/lxswpvdk.html`

- z/VM TCP/IP planning, customization, and reference:

  `http://www.vm.ibm.com/related/tcpip/tcp-pubs.html`
- z/VM TCP/IP cryptographic security:

  `http://www.vm.ibm.com/related/tcpip/vmsslinf.html`
- z/VM user's guides and command references (XEDIT, Conversational Monitor System (CMS), and others):

  `http://www.vm.ibm.com/library/zvmpdf.html`
- XEDIT for VM/SP System Product R3 (historical reference):

  `http://ukcc.uky.edu/ukccinfo/391/xeditref.html`
- Rex Swain's XEDIT summary page

  `https://rexswain.com/xedit.html`
- Debian Linux S/390 port:

  `https://www.debian.org/ports/s390/`

Important z/VM files

z/VM differs from Linux in the location and number of configuration files. In Linux, many configuration files exist and most of them are in or under the `/etc/` directory. On z/VM, relatively few configuration files exist. However, they are on many different minidisks. Table B-1 on page 478 provides a summary and the location of important z/VM configuration files.

Table B-1 Important z/VM configuration files

File	Location	Description
`SYSTEM CONFIG`	`PMAINT CF0`	This file is the operating system's main configuration file. It defines the system name, control program (CP) volumes, user volumes, and other settings.
`USER DIRECT`	`MAINT 2CC`	This file is the initial z/VM user directory. All virtual machines that are known to the system are defined here. If a directory maintenance product is in use, this file is no longer authoritative.
`PROFILE TCPIP`	`TCPMAINT 198`	This file defines the resources for the primary z/VM TCP/IP stack, including the TCP/IP address, Open Systems Adapter (OSA) resources, subnet mask, and gateway. It is initially created by the IPWIZARD tool as `PROFILE TCPIP`.
`SYSTEM DTCPARMS`	`TCPMAINT 198`	This file is created to define the TCP/IP stacks on the system. It is initially created by the IPWIZARD tool.
`TCPIP DATA`	`TCPMAINT 592`	This file defines the Domain Name System (DNS) server, the domain name, and other settings. It is initially created by the IPWIZARD tool.
`PROFILE EXEC`	`AUTOLOG1 191`	This file is a REXX EXEC that is run when the system starts. It is analogous to the `/etc/inittab` file in Linux.

Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors.

XEDIT cheat sheet

XEDIT has line commands that are typed on the command line (====>) and prefix commands, which are typed over the line numbers on the left side of the window.

Line commands

Do not include the angle brackets (< >) in your commands:

a	Add a line.
a<n>	Add <n> lines.
c/<old>/<new>/<n> <m>	Search for string <old> and replace it with <new> for <n> lines below the current line and <m> times on each line. An asterisk (*) can be used for <n> and <m>.
/<string>	Search for ‘string’ from the current line.
-/<string>	Search backwards for ‘string’.
all /<string>/	Show all occurrences of ‘string’ and hide other lines.
bottom	Move to the bottom of the file.
top	Move to the top of the file.
down <n>	Move down ‘n’ lines.
up <n>	Move up ‘n’ lines.
file	Save the current file and exit XEDIT.
ffile	Save the current file and exit but do not warn of overwrite.
save	Save the current file but do not exit.
quit	Exit XEDIT if no changes were made.
qquit	Exit XEIDT even if changes were not saved.
left <n>	Shift ‘n’ characters to the left.
right <n>	Shift ‘n’ characters to the right.
get <file>	Copy file and insert past the current line.
input	Enable INPUT mode to insert multiple lines of text, beginning at the current line.
:<n>	Move to line ‘n’.
?	Display the last command.
=	Execute the last command.
x <file>	Edit ‘file’ and put it into the XEDIT “ring”.
x	Move to the next file in the ring.

Prefix commands

Prefix commands are typed over the line numbers on the left side of the window:

a	Add one line.
a<n>	Add 'n' lines.
c	Copy one line.
cc	Copy a block of lines.
d	Delete one line.
dd	Delete a block of lines.
f	Line after which a copy (c) or a move (m) is to be inserted.
p	Line before which a copy (c) or a move (m) is to be inserted.
i	Insert a line.
i<n>	Insert 'n' lines.
m	Move one line.
mm	Move a block of lines.
"	Replicate a line.
"<n>	Replicate a line 'n' times.
""	Replicate a block of lines.

You may also want to refer to “Online resources” on page 477, for a more expansive XEDIT summary option.

A vi cheat sheet

The following list is a small subset of **vi** commands that are most commonly used. The vi editor has three modes:

- Input mode: The Insert key, **i**, **o** (add a line below), **O** (add a line above), and other commands put you in this mode where you can type text into the file. When you are in this mode, you see the text --INSERT-- in the last line.
- Command mode: The Esc key takes you out of input mode and into command mode. You can issue the following commands:

i	Brings you back to input mode.
dd	Deletes a line and puts it in the buffer.
<n>dd	Delete <n> lines.
x	Delete a character.
dw	Delete a word.
p	Add the buffer past the current location.
P	Add the buffer before the current location.
o	Add a line and go into insert mode.
/string	Search for string.
n	Execute the last command again (This function can be powerful).
jkl;	Cursor movement.
A	Add text at the end of the line.
<nn>G	Go to line <nn>.
G	Go to the last line in the file.
yy	Yank a line (copy into buffer).
<n>yy	Yank n lines.

- Command-line mode: Pressing the colon (:) key brings you to this mode at the bottom of the window. You can issue the following commands:

:wq	Save (write and quit).
:q!	Quit and discard changes.
:<nn>	Go to line number <nn>.
:r <file>	Read <file> into the current file.
:1,$s/old/new/g	Globally replace <old> with <new>.
:help	Give help.

DirMaint cheat sheet

The following list shows useful DirMaint commands:

Add	Add a user or profile directory entry.
AMDisk	Add a minidisk.
DEDicate	Add or delete an existing dedicate statements.
DMDisk	Remove a minidisk.
FILE	Add or replace a DirMaint control file.
RLDCode	Reload the DirMaint resident operating procedures.
RLDExtn	Reload the DirMaint CONFIG* DATADVH file.
REView	Review a user or profile directory entry.
MDisk	Change the access mode and passwords for minidisks.
STorage	Change the logon storage size.
SEND	Request a copy of a DirMaint control file.
SETOptn	Add, change, or delete CP options.
CLAss	Change the CP class for a directory entry.
SPEcial	Add or delete an existing special statement.

DirMaint example commands

The following examples show DirMaint commands:

- Add a new 50 cylinder minidisk 200 to user ID `spiedie`:

```
DIRMAINT FOR SPIEDIE AMDISK 0200 3390 AUTOG 00050 {VOLUMEGROUP}
```

- Add a link statement to the TCPMAINT 592 minidisk into the directory entry for user `vmfrau`:

```
DIRMAINT FOR VMFRAU LINK TCPMAINT 0592 0592 RR
```

Editing a full profile record from DirMaint

```
DIRMAINT FOR SOMEUSER GET LOCK
RECEIVE 0234 = = A
XEDIT SOMEUSER DIRECT A
DIRMAINT FOR SOMEUSER REPLACE
```

Important: Consider the following points:

- While you are editing a directory entry that you received by using the **DIRMAINT FOR ... GET** command, the last line of the file contains internal data that is used by DirMaint during processing.

  Do not change, delete, or move the line beginning with `*DVHOPT`.
- If you accidentally delete or modify the `*DVHOPT` line, use the XEDIT subcommand **QQUIT** to quit without saving your changes, then restart your XEDIT session for the file. This approach will work if you did *not* use the `SAVE` subcommand during your XEDIT session.

  If you performed an intermediate SAVE, use **QQUIT** to exit without saving any further changes, **ERASE** the locally saved directory entry from your A disk, unlock the record by issuing the command **DIRMAINT FOR ... UNLOCK**, and then start over again.

Blank planning worksheet

This section contains a blank copy of the planning worksheet that is used in section 2.1.2, "IBM Dynamic Partition Manager" on page 36. This worksheet is included for your convenience. Hopefully, it is organized in the order that you will need the data. It is recommended that you specify all of the applicable values in the worksheet to simplify and expedite your installation process.

Tip:

Print out these pages that comprise the planning worksheet so you can physically write on them to complete each section. We recommend that you use simplex mode (printing only on one side of each piece of paper). You will have an easier time completing the sheets when you do not have to flip back and forth as much. In addition, if you choose to then scan your completed sheets for reference/archive purposes, this can greatly simplify that task as well.

IBM Shop Z

If you are ordering z/VM by using *Shop Z*, as described in section 5.1, "Obtaining z/VM through electronic download" on page 98, use Table B-2 to record the values that you will use.

Shop Z home page: http://www.ibm.com/software/shopzseries

Table B-2 Shop Z data

Name	Value
User ID	
Order number	
Order name	
Date/Time	

Hardware Management Console

In Chapter 5.3, “Installing z/VM from a DVD or an FTP server” on page 110, we describe how to start a z/VM installation from the Hardware Management Console (HMC). Complete Table B-3 to record the values that you will use.

Table B-3 HMC values

Name	Value
HMC location or URL	
HMC user ID	
FTP source system (if installing from FTP)	
z/VM installation directory	

z/VM Installation Planning Panels (INSTPLAN)

In Chapter 5.4.2, “In-memory z/VM system loaded” on page 113, we describe the INSTPLAN command under step number 2 on page 115 that is run from the Integrated 3270 Console. The following information will be necessary.

INSTPLAN panels 1 and 2

Complete Table B-4 to record the values that are required in the first two INSTPLAN panels.

Table B-4 INSTPLAN values for the first two panels

Name	Value	Comment
Product install target	☐ **F** (SFS filepool) ☐ **M** (minidisk)	Leave set to the default value of **F** for all
Language	☐ AMENG ☐ UCENG	Select AMENG (American English)
DASD model	☐ 3390 **10016** ☐ 3390 ________	***3390 10016*** is a reference to 3390 Model-9. If you are installing to a larger disk size, overtype 10016 with the cylinder count for the disks you will use.
File pool name	**VMPSFS**	VMPSFS (default) recommended.
System type	☐ SSI (VMSSI) ☐ Non-SSI	
Non-SSI system name		Used for non-SSI installation only.
Number of members		VMSSI installation only (usually 2 or 4).
SSI cluster name		VMSSI installation only.
Automatic configuration	NO	

INSTPLAN panel 3

Complete Table B-5 to record the values that are required in the third INSTPLAN panel. The member names will become the z/VM system identifiers, and the logical partition (LPAR) names need to be the same names that are on the HMC.

Table B-5 INSTPLAN values for panel 3

Slot	Member name	LPAR name	Comment
1			Member 1 system identifier and LPAR name
2			Member 2 system identifier and LPAR name
3			Member 3 system ID and LPAR name (optional)
4			Member 4 system ID and LPAR name (optional)

INSTPLAN worksheet 3

Complete Table B-6 to record the volume labels and real device addresses that are required in the Installation Volume Definition INSTPLAN panel.

Table B-6 INSTPLAN values worksheet for volume definition

Type	Default label	Chosen label	Address	Comment
COMMON	VMCOM1			Common volume 1
COMMON2	VMCOM2			Common volume 2
RELVOL	630RL1			Release volume 1
RELVOL2	630RL2			Release volume 2
Mem 1 RES	M01R01			Member 1 residence volume
Mem 1 SPOOL	M01S01			Member 1 spool volume
Mem 1 PAGE	M01P01			Member 1 page volume
Mem 1 WORK	M01W01			Member 1 work volume 1
Mem 1 WORK	M01W02			Member 1 work vol 2 (3390-3 only)
Mem 1 WORK	M01W03			Member 1 work vol 3 (3390-3 only)
Mem 2 RES				Member 2 residence volume
Mem 2 SPOOL				Member 2 spool volume
Mem 2 PAGE				Member 2 page volume
Mem 2 WORK				Member 2 work volume 1
Mem 2 WORK				Member 2 work vol 2 (3390-3 only)
Mem 2 WORK				Member 2 work vol 3 (3390-3 only)
Mem 3 RES				Member 3 residence vol (optional)
Mem 3 SPOOL				Member 3 spool volume
Mem 3 PAGE				Member 3 page volume
Mem 3 WORK				Member 3 work volume 1

Type	Default label	Chosen label	Address	Comment
Mem 3 WORK				Member 3 work vol 2 (3390-3 only)
Mem 3 WORK				Member 3 work vol 3 (3390-3 only)
Mem 4 RES				Member 4 residence vol (optional)
Mem 4 SPOOL				Member 4 spool volume
Mem 4 PAGE				Member 4 page volume
Mem 4 WORK				Member 4 work volume 1
Mem 4 WORK				Member 4 work vol 2 (3390-3 only)
Mem 4 WORK				Member 4 work vol 3 (3390-3 only)

INSTPLAN worksheet 4

Complete the worksheet in Table B-7 to record the common volume and channel-to-channel (CTC) addresses that are required in the INSTPLAN panel. This panel is shown at the end of 5.6.1, “Copying in-memory z/VM system to DASD” on page 124.

If only two members exist in the SSI, you need to specify only two pairs of CTCs (from member 1 to member 2, and vice versa).

Table B-7 INSTPLAN values worksheet for volume definition

Real addresses for the common volume on each member LPAR:			
Member 1	**Member 2**	**Member 3**	**Member 4**
CTC device addresses:			
From member 1		**From member 2**	
To: member 1	N/A	To: member 1	_______ _______
To: member 2	_______ _______	To: member 2	N/A
To: member 3	_______ _______	To: member 3	_______ _______
To: member 4	_______ _______	To: member 4	_______ _______
From member 3		**From member 4**	
To: member 1	_______ _______	To: member 1	_______ _______
To: member 2	_______ _______	To: member 2	_______ _______
To: member 3	N/A	To: member 3	_______ _______
To: member 4	_______ _______	To: member 4	N/A

z/VM networking resources

Complete the worksheet in Table B-8 to list the networking resources that will be needed when you start the **IPWIZARD** and when you create a VSWITCH for the Linux virtual machines.

Table B-8 z/VM and networking resources worksheet

Name	Value	Comment
TCP/IP user ID		TCPIP is recommended.
z/VM host name, member 1		
z/VM host name, member 2		
TCP/IP domain name		System domain name is usually set in DNS.
TCP/IP gateway		The router to and from the local subnet.
DNS server 1		Assigned by the network administrator.
DNS server 2/3		Optional.
Interface name		
OSA starting device number		Start of OSA *triplet* for z/VM TCP/IP stack.
Subnet mask		Assigned by network administrator.
OSA device type		
Maximum transmission unit (MTU) size		Check with network administrator.
Primary OSA device for virtual switch		Specify the first real device number and the next two device numbers will also be used.
Secondary OSA device for virtual switch		Ideally, it needs to be on a different channel-path identifier (CHPID)/OSA.

z/VM DASD worksheet

Use the worksheet in Table B-9 to document the z/VM DASD that you will use.

Table B-9 z/VM DASD blank worksheet

Device number	Label	Type	Notes

Device number	Label	Type	Notes

Linux resources worksheet

Use the worksheet in Table B-10 to document the resources that are associated with the Network File System (NFS) server that will be used to be the installation source of the first Linux on z Systems.

Table B-10 Linux NFS server resources blank worksheet

Name	Value	Comment
TCP/IP address		
User/password		
NFS-exported installation directory		

Use the worksheet in Table B-11 to document your Linux on z Systems resources.

Table B-11 Linux resources blank worksheet

Name	Value	Comment
Linux installation password		
Linux root password		
Linux TCP/IP gateway		
Linux TCP/IP broadcast		
Linux DNS server		
Virtual Network Computing (VNC) installation password		

Host names and IP addresses worksheet

Use the worksheet in Table B-12 to document the host names and associated IP addresses and virtual machines that you will use.

Table B-12 Host names blank worksheet

Host name	IP address	Virtual machine/ LPAR	Notes

C

Additional material

This book refers to additional material that can be downloaded from the internet. It includes the following topics:

Locating the web material

The web material that is associated with this book is available on the Internet. You can obtain this material at the following URL:

`http://ibm.com/vm/pubs/redbooks/sg248147`

Using the web material

The files that are associated with this book are in a *GNU* compressed tar file.

The additional web materials that accompany this book are in the following file:

File name	*Description*
SG248147.tgz	Code samples in compressed tar format

Within the tar file, the directory `SG248147/` contains the following subdirectories and files:

`disclaimer.txt`	Legal disclaimer
`README.txt`	Description file
`rhel64/`	Directory with files for RHEL 6.4
`rhel64/clone-1.0-11.s390x.rpm`	RHEL 6.4 clone RPM
`sles11sp3/`	Directory with files for SLES 11 SP3
`sles11sp3/clone.sh`	SLES 11 SP3 clone script
`sles11sp3/linux5.xml`	AutoYaST profile
`sles11sp3/boot.clone`	Init script for new clones
`sles11sp3/jeos.tgz`	Files that are associated with kiwi
`vm/`	Directory with files for z/VM
`vm/lnxmaint/`	Directory with files for LNXMAINT 192
`vm/lnxmaint/rhel64.exec`	EXEC to start an RHEL 6.4 installation
`vm/lnxmaint/sample.parm-rh6`	Sample RHEL 6.4 parameter file
`vm/lnxmaint/sample.conf-rh6`	Sample RHEL 6.4 configuration file
`vm/lnxmaint/sample.parm-s11`	Sample SLES 11 SP3 parameter file
`vm/lnxmaint/profile.exec`	Sample PROFILE EXEC for Linux IDs
`vm/lnxmaint/sles11s3.exec`	EXEC to start an SLES 11 SP3 installation
`vm/lnxmaint/swapgen.exec`	EXEC to define VDISK swap spaces
`vm/maint/`	Directory with files for MAINT 191
`vm/maint/callsm1.exec`	EXEC to test Systems Management API (SMAPI)
`vm/maint/cpformat.exec`	EXEC to format multiple DASD volumes
`vm/maint/ssicmd.exec`	EXEC to run a command on all single system image (SSI) members

System requirements for downloading the web material

The web material requires the following system configuration:

Hard disk space: 41 KB
Operating System: Linux

Downloading and extracting the web material

This section lists code that is associated with this book.

z/VM REXX EXECs and XEDIT macros

This section lists all of the z/VM code that is included in the associated tar file:

- CPFORMAT EXEC
- SSICMD EXEC
- PROFILE EXEC for Linux virtual machines
- RHEL64 EXEC
- SLES11S3 EXEC

CPFORMAT EXEC

The following code is for the EXEC that formats multiple DASD using **CPFMTXA**. It is described in 6.12, "Enabling z/VM basic system automation" on page 196.

```
/***********************************************************/
/*                                                         */
/*  This program is provided on an "AS IS" basis, without  */
/*  warranties or conditions of any kind, either express or */
/*  implied including, without limitation, any warranties   */
/*  or conditions of title, non-infringement,               */
/*  merchantability or fitness for a particular purpose.    */
/*  Neither recipient nor any contributors shall have any   */
/*  liability for any direct, indirect, incidental,         */
/*  special, exemplary, or consequential damages (including */
/*  without limitation lost profits), however caused and on */
/*  any theory of liability, whether in contract, strict    */
/*  liability, or tort (including negligence or otherwise)  */
/*  arising in any way out of the use or distribution of    */
/*  the program or the exercise of any rights granted       */
/*  hereunder, even if advised of the possibility of such   */
/*  damages.                                                */
/*                                                         */
/***********************************************************/
/*                                                         */
/* Purpose:                                                */
/*   CP format one, a range or multiple ranges of DASD.    */
/*   and label these DASDs.                                */
/*                                                         */
/* Inputs:                                                 */
/*   dasds - address(es) of the DASD to format.            */
/*   type  - type of formatting to be done: PERM, PAGE, SPOL */
/*           or TEMP.                                      */
/*                                                         */
/* Output:                                                 */
/*   Virtual DASD that is CP formatted and labeled.        */
/*                                                         */
/* Return codes:                                           */
/*   0 - success                                           */
/*   1 - help was asked for or given                       */
```

```
/*   2 - user did not respond Y to confirm formatting        */
/*   3 - DASD (minidisk) range is not valid                  */
/*   4 - at least one DASD (minidisk) is reserved to MAINT   */
/*                                                           */
/* References:                                               */
/*  The Cloud Computing Cookbook for z/VM 6.2, RHEL 6.2 and  */
/*    SLES 11 SP2                                            */
/*  URL: http://www.vm.ibm.com/devpages/mikemac/SG248147.pdf */
/*                                                           */
/***********************************************************/
  Address COMMAND
  firstchar = 'J'
  Arg dasds 'AS ' type .
  If dasds = '' | dasds = '?' Then Call help
  labelPrefix = firstchar || getLabelPrefix(type)
  numDasd = parseDasd(dasds)
  answer = areYouSure(type)
  If answer = 'Y' Then Do
     /* the user is sure */
     formatted = ''
     retVal = doFormat(labelPrefix numDasd type)
     Call doReport retVal
     End
  Else retVal = 2
  Exit retVal

/*+------------------------------------------------------------------+*/
help:
  Procedure Expose firstchar
/*+------------------------------------------------------------------+*/
  Parse Source . . fn .
  Say
  Say 'Synopsis:'
  Say
  Say '  Format and label DASD as page, perm, spool or temp disk space'
  Say '  The label written to each DASD is' firstchar || '<t><xxxx> where:'
  Say '    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)'
  Say '    <xxxx> is the 4 digit address'
  Say
  Say 'Syntax is:'
  Say "                <---------------<              "
  Say "   >>--CPFORMAT--.-vdev--------.--AS---.-PERM-.--------->< "
  Say "                 '-vdev1-vdev2-'       '-PAGE-'"
  Say "                                       '-SPOL-'"
  Say "                                       '-TEMP-'"
  Say
  Exit 1

/*+------------------------------------------------------------------+*/
areYouSure:
  Procedure
/*| Warn the user of possible data loss and ask if it is okay to     |*/
/*|  format the DASD.                                                |*/
/*|  parm 1: format type for the virtual DASD                        |*/
/*|  retVal: first character of response. continue if 'Y'.          |*/
```

```
/*+------------------------------------------------------------------+*/
  Arg type
  Say
  Say 'WARNING - this will destroy data!'
  Say 'Are you sure you want to format the DASD as' type 'space (y/n)?'
  Pull answer .
  Return  'LEFT'(answer,1) /* from areYouSure */

/*+------------------------------------------------------------------+*/
getLabelPrefix:
  Procedure expose firstchar
/*| Return the second character of the virtual DASD label            |*/
/*|  parm 1: format type for the virtual DASD                        |*/
/*+------------------------------------------------------------------+*/
  Arg type .
  firstchar. = 0
  firstchar.PERM = 'M'
  firstchar.PAGE = 'P'
  firstchar.SPOL = 'S'
  firstchar.TEMP = 'T'
  If firstchar.type = 0 Then Do
  /* Incorrect formatting type specified. Provide help and quit. */
     Say 'Error: "AS" must be present, type must be PERM, PAGE, SPOL or TEMP'
     Call help
     End
  Return firstchar.type

/*+------------------------------------------------------------------+*/
parseDASD:
  Procedure Expose dasdList.
/*| parse all dasd into an array verifying all are attached          |*/
/*|  parm 1: dasds - the list of dasd passed in                      |*/
/*|  retVal: number of DASD in dasdList                              |*/
/*+------------------------------------------------------------------+*/
  Arg dasds
  numDasd = 0
  dropheader = ''
  Say
  Say 'Format the following DASD:'
  Do While dasds <> ''
     Parse Upper Var dasds dasd dasds
     dashPos = 'POS'('-',dasd)
     If dashPos = 0 Then Do
     /* There is a singleton DASD specified. */
     /* start and end of range are the same. */
        startrange = dasd
        endrange = dasd
        End
     /* process the range of DASD */
     Else Parse Var dasd startrange  '-' endrange
     Do i = 'X2D'(startrange) To 'X2D'(endrange)
        numDasd = numDasd + 1
        dasdList.numDasd = 'D2X'(i)
        'PIPE CP QUERY MDISK' dasdList.numDasd 'LOCATION',
        dropheader,
```

```
      '|CONS'
      If rc <> 0 Then Do
         Say 'Return code from QUERY MDISK =' rc
         /* If RC=40, then HCPxxx40E has been issued and msg below */
         If rc = 40 Then Say 'DASD' dasdList.numDasd 'is not attached.'
         Exit 3
         End
      Call checkReserved(dasdList.numDasd)
      dropheader = '|DROP 1'
      End
   End
 Return numDasd /* from parseDasd */

/*+------------------------------------------------------------------+*/
doFormat:
 Procedure Expose dasdList. formatted
/*| Format all DASD specified using CPFMTXA                          |*/
/*|  parm 1: labelPrefix - the two character label prefix           |*/
/*|  parm 2: numDasd - number of DASD in the array dasdList         |*/
/*|  parm 3: type - the type of DASD format                         |*/
/*|  retVal: 0 = success                                            |*/
/*+------------------------------------------------------------------+*/
 Arg labelPrefix numDasd type
 /* Save the current settings for MORE */
 Parse Value 'DIAG'('08','CP QUERY TERM') With ' MORE' morevalues ','
 'CP TERM MORE 1 1' /* Make MORE brief */

 /* Save system identifier and SSI name */
 'PIPE CP QUERY USERID | SPEC W3 | VAR systemID'
 'PIPE CP QUERY SSI | LOCATE /SSI Name/ | SPEC W3 | VAR SSIname'
 If (SSIname = "SSINAME") Then /* variable not set */
   inSSI = 'no'
 Else
   inSSI = 'yes'

 /* Iterate through all DASD in list */
 Do i = 1 to numDasd
    label = labelPrefix  || 'RIGHT'(dasdList.i,4,'0')
    retVal = formatOne(dasdList.i type label)
    If retVal <> 0 Then Do
       Say 'Error from CPFMTXA on DASD' label 'rc =' retVal
       Leave /* error - abort this format */
       End

    /* add owner info for CP owned devices */
    If (type != 'PERM') Then /* CP owned => owner info is needed */
      If (inSSI = 'yes') Then /* add owner info */
        call addOwnerInfo(dasdList.i label SSIname systemID)
      Else
        call addOwnerInfo(dasdList.i label "NOSSI" systemID)
    formatted = formatted label
    End /* Do i = */
 'CP TERM MORE' morevalues
 Return retVal /* from doFormat */
```

```
/*+------------------------------------------------------------------+*/
checkReserved:
  Procedure
/*| Try copying an already formatted DASD Then relabelling it        |*/
/*|  parm 1: dasd - the virtual address of the DASD                  |*/
/*+------------------------------------------------------------------+*/
  Arg dasd
  /* Create a list of reserved virtual DASD addresses. */
  /* Ensure that a system minidisk is not formatted.   */
  resvd = '122 123 124 190 191 193 19D 19E 2CC 401 402 990 CF1 CF3 CFD'
  If 'POS'(resvd,dasd) <> 0 Then Do
     /* MAINT minidisk - ABORT! */
     Say 'Minidisk' dasd 'is a reserved MAINT minidisk'
     Say 'This must be formatted manually using a different vaddr.'
     Exit 4
     End /* If dasd is reserved */
  Return /* from checkReserved */

/*+------------------------------------------------------------------+*/
doReport:
  Procedure Expose dasds formatted
/*| Report on the newly labelled DASD                                |*/
/*|  parm 1: formatSuccess - 0=all is well, non-0= a format failed   |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  Arg formatSuccess
  If formatSuccess <> 0 Then
    Say 'Error was encountered! retVal from CPFMTXA =' formatSuccess
  If formatted = '' Then
    Say 'No DASD were successfully formatted'
  Else
    Say 'DASD successfully formatted:' formatted
  'CP DETACH' dasds
  'CP ATTACH' dasds '*'
  Say
  Say 'DASD status after:'
  'CP QUERY MDISK' dasds 'LOCATION'
  Return 0 /* from doReport */

/*+------------------------------------------------------------------+*/
formatOne:
  Procedure
/*| Format a DASD via DDR                                            |*/
/*|  parm 1: disk - the vaddr to be formatted                        |*/
/*|  parm 2: type - PERM, PAGE, SPOL or TEMP                         |*/
/*|  parm 3: label - the six character label                         |*/
/*+------------------------------------------------------------------+*/
  Arg disk type label
  Queue 'FORMAT'
  Queue disk
  Queue '0 END'
  Queue label
  Queue 'YES'
  Queue type '0 END'
  Queue 'END'
```

```
  'EXEC CPFMTXA'
  retVal = rc
  Return retVal /* from formatOne */

/*+------------------------------------------------------------------+*/
AddOwnerInfo:
  Procedure
/*| Tag PAGE, SPOL and TDSK volumes with SSI                        |*/
/*|  parm 1: disk - the vaddr to be formatted                        |*/
/*|  parm 2: type - PERM, PAGE, SPOL or TEMP                         |*/
/*|  parm 3: label - the six character label                         |*/
/*+------------------------------------------------------------------+*/
  Arg disk label SSIname systemID
  Queue 'OWNER'
  Queue disk
  Queue label
  Queue SSIname
  Queue systemID
  'EXEC CPFMTXA'
  retVal = rc
  Return retVal /* from addOwnerInfo */
```

SSICMD EXEC

The following code is for the EXEC that issues control program (CP) commands on all joined members of a single system image (SSI) cluster. It is recommended to be on the `MAINT 191` disk.

```
/***********************************************************/
/*                                                         */
/*  This program is provided on an "AS IS" basis, without    */
/*  warranties or conditions of any kind, either express or  */
/*  implied including, without limitation, any warranties    */
/*  or conditions of title, non-infringement,                */
/*  merchantability or fitness for a particular purpose.     */
/*  Neither recipient nor any contributors shall have any    */
/*  liability for any direct, indirect, incidental,          */
/*  special, exemplary, or consequential damages (including  */
/*  without limitation lost profits), however caused and on  */
/*  any theory of liability, whether in contract, strict     */
/*  liability, or tort (including negligence or otherwise)   */
/*  arising in any way out of the use or distribution of     */
/*  the program or the exercise of any rights granted        */
/*  hereunder, even if advised of the possibility of such    */
/*  damages.                                                 */
/*                                                         */
/***********************************************************/
/*                                                         */
/* Purpose:                                                  */
/*   Issue a command on all members of a cluster using the   */
/*   response from QUERY SSI to find the member names.       */
/*                                                         */
/* Inputs:                                                   */
/*   cmd - the CP command to issue on each member.           */
/*                                                         */
```

```
/* Output:                                                  */
/*   The results from issuing the AT command.               */
/*                                                          */
/* References:                                              */
/*  The Cloud Computing Cookbook for z/VM 6.2, RHEL 6.2 and */
/*    SLES 11 SP2                                           */
/*  URL: http://www.vm.ibm.com/devpages/mikemac/SG248147.pdf */
/*                                                          */
/**********************************************************/
  Address COMMAND
/* The command is passed by the caller */
  Arg cmd
  /* Provide help if requested or if no command is specified */
  If cmd = '' | cmd = '?' Then Call Help
  /* Determine the members of the SSI cluster */
  'PIPE CP QUERY SSI',
  '| STEM MSG.',           /* Save the response if error */
  '| XLATE',               /* Make all output upper case */
  '| FRTARGET ALL /SLOT/', /* Just look after 'SLOT' */
  '| LOCATE /JOINED/',     /* JOINED members can do a command */
  '| SPEC W2',             /* Get the member names */
  '| STEM SSI.'            /* Save the member names */
  /* If nonzero return code, show error message and exit */
  If rc <> 0 | ssi.0 = 0 Then Do
     Say 'Error: QUERY SSI return code =' rc
     Say msg.1
     End
  Else Do
  /* Send the command to each member of the SSI cluster */
     Do i = 1 To ssi.0
        Say ssi.i||":"
        'CP AT' ssi.i 'CMD' cmd
        Say
        End
     End
  Exit

help:
  /* Provide syntax information to the user */
  Say 'SSICMD cmd'
  Say
  Say 'cmd is a command to be issued on each of the members'
  Say '  in the SSI cluster using the AT command.'
  Exit
```

PROFILE EXEC for Linux virtual machines

This section lists the code for the `PROFILE EXEC` that is shared among Linux virtual machines from the `LNXMAINT 192` disk.

```
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
```

```
'SWAPGEN 300 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else  /* user is interactive -> prompt */
 do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
 end
```

RHEL64 EXEC

This section lists the code for the `RHEL64 EXEC` that starts an RHEL 6.4 installation. It is recommended to be on the `LNXMAINT 192` disk.

```
/************************************************************/
/* Punch a RHEL 6.4 install system to reader and IPL it      */
/* Input files: RHEL64 KERNEL, <ID> PARM-RH6, RHEL64 INITRD  */
/************************************************************/
  Address 'COMMAND'
  'CP SPOOL PUN *'
  'CP CLOSE RDR'
  'CP PURGE RDR ALL'
  'PUNCH RHEL64 KERNEL * (NOHEADER'
  'PUNCH' 'USERID'() 'PARM-RH6 * (NOHEADER'
  'PUNCH RHEL64 INITRD * (NOHEADER'
  'CP CHANGE RDR ALL KEEP'
  'CP IPL 00C CLEAR'
  Exit
```

SLES11S3 EXEC

This section lists the code for the `sles11s3 EXEC` that starts a SLES 11 SP3 installation. It is recommended to be on the `LNXMAINT 192` disk.

```
/* Punch a SLES 11 SP3 install system to reader and IPL it   */
/************************************************************/
  Address 'COMMAND'
  'CP SPOOL PUN *'
  'CP CLOSE RDR'
  'CP PURGE RDR ALL'
  'PUNCH SLES11S3 KERNEL * (NOHEADER'
  'PUNCH' 'USERID'() 'PARM-S11 * (NOHEADER'
  'PUNCH SLES11S3 INITRD * (NOHEADER'
  'CP CHANGE RDR ALL KEEP'
  'CP IPL 00C CLEAR'
  Exit
```

SWAPGEN EXEC

The commands in this EXEC create a Linux swap partition on VM disks. The following code is for the EXEC that creates Linux swap spaces from z/VM VDISKs. You can download this EXEC from this web page.

Sample files

This section lists the sample files that are described in the book.

SAMPLE CONF-RH6 file

This section lists the sample RHEL 6 configuration file.

```
DASD=100-103,300-301
HOSTNAME=hostName.DNSname.com
NETTYPE=qeth
IPADDR=n.n.n.n
SUBCHANNELS=0.0.0700,0.0.0701,0.0.0702
NETMASK=255.255.255.0
SEARCHDNS=DNSname.com
GATEWAY=n.n.n.n
DNS=n.n.n.n
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
```

SAMPLE PARM-RH6 file

This section lists the sample RHEL 6 configuration file.

```
root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=userid.CONF-RH6
vnc vncpassword=12345678
```

SAMPLE PARM-S11 file

This section lists the sample SLES 11 SP3 configuration file.

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=n.n.n.n Hostname=yourhost.example..com
Gateway=n.n.n.n Netmask=255.255.255.0
Broadcast=n.n.n.n Layer2=1
ReadChannel=0.0.0700  WriteChannel=0.0.0701  DataChannel=0.0.0702
Nameserver=n.n.n.n
portname=whatever
portno=0
Install=nfs://n.n.n.n/var/nfs/sles11sp3/SLES-11-SP2-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

Linux code

This section contains listings of the following Linux scripts:

- RHEL clone script
- SLES clone.sh script
- SLES boot.clone script

RHEL clone script

This section lists the code for the `/usr/sbin/clone` script that clones from an RHEL golden Linux image to a target virtual machine. It is contained in the RPM `clone-1.0-11.s390x.rpm`.

```
#!/bin/sh
#
# clone.sh is a script that clones Linux images.  It makes use of vmcp to
# relay messages to the z/VM system and configuration files to modify
# the new image once it has been cloned.
#
# The script reads in /etc/sysconfig/clone for user setting customizations.
#
# For details on how this script works see the book:
#   "z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL4"
#   on the Web at: http://www.redbooks.ibm.com/abstracts/sg247272.html
#
# ----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# ----------------------------------------------------------------------------

# These MUST be lower case!
MASTER_LINK=fffe
CLONE_LINK=ffff

#+--------------------------------------------------------------------------+
function help
# give help
#+--------------------------------------------------------------------------+
 {
  echo "Usage: clone [-v] sourceID targetID [rootMinidisk [minidisk1
minidisk2..]]"
  echo "    Switches"
  echo "       -v Verbose output"
  echo "    Required"
  echo "       sourceID the z/VM user id you want to clone from"
  echo "       targetID  the z/VM user id you want to clone to"
```

```
  echo "    Optional"
  echo "       rootMinidisk the minidisk address that contains the root
filesystem"
  echo "       minidisk1..n additional minidisks that should be copied"
  exit
 }

#+--------------------------------------------------------------------------+
function cp_cmd
# echo a CP command and invoke it via cp_cmd
#   Arg1-n: the z/VM command to issue
#   Return: the z/VM command's return code
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Invoking CP command: $@"
  out=$(vmcp $@ 2>&1)
  rc=$?

  # Pull the z/VM error code from the output
  if [ $rc -ne 0 ] ; then
   rc=$(echo $out | grep Error | sed s/.*#//g)
   [ -z "$rc" ] && rc=1
  fi
  return $rc
 }

#+--------------------------------------------------------------------------+
function copy_key
# If the host has a id_dsa.pub file then append that to the clone's
# authorized_keys file.
#+--------------------------------------------------------------------------+
 {
  if [ -e /root/.ssh/id_dsa.pub ] ; then
    [ ! -d /mnt/clone/root/.ssh/ ] && mkdir -p /mnt/clone/root/.ssh/
    echo "# LNXINST" >> /mnt/clone/root/.ssh/authorized_keys
    cat /root/.ssh/id_dsa.pub >> /mnt/clone/root/.ssh/authorized_keys
    chmod 600 /mnt/clone/root/.ssh/authorized_keys
  fi
 }

#+--------------------------------------------------------------------------+
function abort
# Exit the script and clean up
#+--------------------------------------------------------------------------+
 {
  umount_cloned_image

  set_offline $CLONE_LINK
  set_offline $MASTER_LINK

  unlink_one $CLONE_LINK
  unlink_one $MASTER_LINK

  exit $1
 }
```

```
#+--------------------------------------------------------------------------+
function get_target_info
# Get the TCP/IP and DNS info for the Linux ID to clone to.  This function
# will check both the shared.conf file and the specific target id's conf
# file.  If values are still missing then the user will be prompted to
# supply them.
#+--------------------------------------------------------------------------+
 {
  unset HOSTNAME
  [ -f /etc/clone/shared.conf ] && . /etc/clone/shared.conf
  [ -f /etc/clone/${target_linux_id}.conf ] && .
/etc/clone/${target_linux_id}.conf

  shift # drop the MasterGuestID
  shift # drop the CloneGuestID

  # If there are still command line arguments then the user must have specified
DASD
  # on the command line.  Unset whatever we have in DASD (from the config files)
and
  # set DASD equal to the rest of the arguments.
  [ $# -gt 0 ] && DASD="$@" && unset DASD_ROOT

  # Loop through all of the values that we require and double check that they have
  # values.  If they don't then we will prompt the user to fill them in.
  for v in HOSTNAME IPADDR DNS GATEWAY NETMASK MTU SUBCHANNELS SEARCHDNS NETTYPE
DASD
  do
   if [ -z "$(eval echo \$$v)" ]; then
    [ "$PROMPT" != "y" ] && echo "Error: missing required value for $v" && exit 1
    [ -z "$first" ] && echo "Please enter $target_linux_id's value for: " &&
first=1
    echo -n "$v: "
    read in
    eval $(echo $v=\"$in\")
    export $v
    echo "$v=$in" >> /etc/clone/${target_linux_id}.conf
   fi
  done

  # Expand DASD ranges if they have been defined
  if [ -n "$DASD" ] ; then
   split=$(echo $DASD | tr ',' ' ')
   DASD=""
   for s in $split
   do
    out=$(echo $s | grep \-)
    rc=$?
    [ $rc -eq 0 ] && DASD=${DASD}$(seq -s" " $(echo $s | tr '-' ' ' | tr '\n' '
'))
    [ $rc -ne 0 ] && DASD=${DASD}$(echo -n "$s ")
   done
   [ -n "$DASD_ROOT" ] && DASD=$(echo $DASD | sed "s/$DASD_ROOT//")
   DASD="$DASD_ROOT $DASD"
```

```
   # Assuming that if no DASD_ROOT is specified then the first DASD device will be
   # take as root
   if [ -z "$DASD_ROOT" ] ; then
     DASD_ROOT=$(echo $DASD | awk -F" " '{print $1}')
   fi
   export DASD
  fi

  # Grab just the hostname with out any DNS suffixes from the FQDN
  target_host=$(echo $target_fqhost | awk -F. '{print $1}')
 }


#+--------------------------------------------------------------------------+
function dd_copy
# Use the dd command to copy one disk to another
#   Arg 1: Source minidisk - assumed to be online
#   Arg 2: Target minidisk - must be brought online and dasdfmt'd
#+--------------------------------------------------------------------------+
 {
  ret_val=0

  source_mdisk=$1
  target_mdisk=$2

  # Bring the source and target devices online
  set_online $source_mdisk
  set_online $target_mdisk

  target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" | awk '{
print $7 }'`
  source_dev_node=`cat /proc/dasd/devices | grep "$source_mdisk(ECKD)" | awk '{
print $7 }'`

  wait_for_device /dev/$target_dev_node
  ret_val=$?

  if [ $ret_val -eq 0 ] ; then
    [ -n "$VERBOSE" ] && echo "Invoking Linux command: dasdfmt -p -b 4096 -y -F -f
/dev/$target_dev_node"
    [ -n "$VERBOSE" ] && progress="-p"
    dasdfmt $progress -b 4096 -y -F -f /dev/$target_dev_node
    [ $? -ne 0 ] && echo "Error: dasdfmt failed" && ret_val=1
  fi

  if [ $ret_val -eq 0 ] ; then
    wait_for_device /dev/$source_dev_node
    ret_val=$?
  fi

  if [ $ret_val -eq 0 ] ; then
    nblks=`cat /proc/dasd/devices | grep $target_dev_node | awk '{ print $13 }'`
    [ -n "$VERBOSE" ] && \
    echo "Invoking Linux command: dd bs=4096 count=$nblks if=/dev/$source_dev_node
of=/dev/$target_dev_node"
```

```
    dd bs=4096 count=$nblks if=/dev/$source_dev_node of=/dev/$target_dev_node
>/dev/null
    [ $? -ne 0 ] && echo "Error: dd failed" && ret_val=1
  fi

  # Put the source and target devices offline
  set_offline $target_mdisk
  set_offline $source_mdisk

  return $ret_val
 }

#+--------------------------------------------------------------------------+
function link_one
# This will link one minidisk from another user id as the target minidisk
# address on the current z/VM user id with a link mode indicated by the
# 4th argument.
#
#   Arg1: Source z/VM ID
#   Arg2: Source minidisk virtual address
#   Arg3: Target minidisk virtual address
#   Arg4: Link mode (rr/w)
#+--------------------------------------------------------------------------+
 {
  source_id=$1
  source_mdisk=$2
  target_mdisk=$3
  link_mode=$4

  cp_cmd QUERY VIRTUAL $target_mdisk
  if [ $? != 40 ]; then
    cp_cmd DETACH $target_mdisk
  fi

  cp_cmd LINK $source_id $source_mdisk $target_mdisk $link_mode $LINK_PASSWD
  if [ $? != 0 ]; then
    echo "cp_cmd link $source_id $source_mdisk $target_mdisk $link_mode failed -
exiting"
    abort 1
  fi
  }

#+--------------------------------------------------------------------------+
function unlink_one
# This will unlink a minidisk from the current z/VM user id.
#   Arg1: The target minidisk to unlink
#+--------------------------------------------------------------------------+
 {
  cp_cmd DETACH $1
  return $?
 }

#+--------------------------------------------------------------------------+
function copy_one
# Try to use z/VM FLASHCOPY to copy one disk to another. If that fails,
```

```
#    call dd_copy() to fall back to the Linux DD command
#   Arg 1: Source minidisk
#   Arg 2: Target minidisk
#+--------------------------------------------------------------------------+
 {
  source_mdisk=$1
  target_mdisk=$2

  if [ "$CLONE_METHOD" == "AUTO" -o "$CLONE_METHOD" == "auto" ] ; then
    cp_cmd FLASHCOPY $source_mdisk 0 END $target_mdisk 0 END
    rc=$?
    if [ $rc -ne 0 ]; then # FLASHCOPY failed
      [ -n "$VERBOSE" ] && echo "FLASHCOPY $source_mdisk $target_mdisk failed with
$rc - using Linux dd"
    else
     return 0
    fi
  fi

  dd_copy $source_mdisk $target_mdisk
  [ $? -ne 0 ] && return 1
 }

#+--------------------------------------------------------------------------+
function copy_disks
# Call copy_one to copy each disk passed in as an argument.
#   Arg1-n: The minidisk address to copy
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Copying minidisks..."
  while [ $# -gt 0 ]; do
    link_one $source_linux_id $1 $MASTER_LINK RR
    link_one $target_linux_id $1 $CLONE_LINK W
    copy_one $MASTER_LINK $CLONE_LINK
    [ $? -eq 0 ] && echo "$1 disk copied ..."
    unlink_one $MASTER_LINK
    unlink_one $CLONE_LINK
    shift
  done
 }

#+--------------------------------------------------------------------------+
function link_disks
# Call link_one to link each disk passed in as an argument.
#   Arg1-n: The minidisk address to link
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Linking minidisks for LVM..."
  while [ $# -gt 0 ]; do
    link_one $target_linux_id $1 400$# W
    set_online 400$#
    [ $? -eq 0 ] && echo "$1 disk linked ..."
    shift
  done
 }
```

```
#+--------------------------------------------------------------------------+
function unlink_disks
# Call unlink_one to unlink each disk passed in as an argument.
#   Arg1-n: The minidisk address to unlink
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Unlinking minidisks ..."
  while [ $# -gt 0 ]; do
    set_offline 400$#
    unlink_one 400$#
    [ $? -eq 0 ] && echo "$1 disk unlinked ..."
    shift
  done
 }

#+--------------------------------------------------------------------------+
function ask_are_you_sure
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "This will copy disks from $source_linux_id to $target_linux_id"
  echo "Host name will be:  $HOSTNAME"
  echo "IP address will be: $IPADDR"
  echo -n "Do you want to continue? (y/n): "
  read ans
  if [ $ans != "y" ]; then
    abort 1
  fi
 }

#+--------------------------------------------------------------------------+
function check_logged_off
# Verify the user ID exists and is logged off
#   Arg1: The user id to query if it is logged on or not
#+--------------------------------------------------------------------------+
 {
  cp_cmd QUERY $1
  case $? in
    0)  # user ID is logged on or disconnected
      echo "$1 user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$1 user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "$1 user ID must exist and be logged off"
      exit 4
  esac
 }
```

```
#+--------------------------------------------------------------------------+
function modify_cloned_image
# Modify the networking information in appropriate files under /etc
# Regenerate SSH keys in golden image's /etc/ssh/ directory and change root pw
#+--------------------------------------------------------------------------+
 {
  source_ipaddr=$(grep IPADDR
$CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0 \
        | awk -F= '{print $2}')
  source_hostname=$(grep HOSTNAME $CLONE_MNT_PT/etc/sysconfig/network \
        | awk -F= '{print $2}')
  source_host=$(echo $source_hostname| awk -F. '{print $1}')

  [ ! -d $CLONE_MNT_PT/etc ] && echo "Error: no $CLONE_MNT_PT/etc found" && abort
1

  [ -n "$VERBOSE" ] && echo "Modifying networking info under $CLONE_MNT_PT..."
  sed -i \
    -e "s/$source_ipaddr/$IPADDR/g" \
    -e "s/$source_hostname/$HOSTNAME/g" \
    -e "s/$source_host/$target_host/g" \
  $CLONE_MNT_PT/etc/hosts

  sed -i \
    -e "s/HOSTNAME=.*/HOSTNAME=$HOSTNAME/g"\
    -e "s/GATEWAY=.*/GATEWAY=$GATEWAY/g"\
  $CLONE_MNT_PT/etc/sysconfig/network

  sed -i \
    -e "s/IPADDR=.*/IPADDR=$IPADDR/g"\
    -e "s/MTU=.*/MTU=$MTU/g"\
    -e "s/NETMASK=.*/NETMASK=$NETMASK/g"\
    -e "s/SUBCHANNELS=.*/SUBCHANNELS=$SUBCHANNELS/g"\
    -e "s/NETTYPE=.*/NETTYPE=$NETTYPE/g"\
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  # Modify MACADDR/HWADDR if specified (optional)
  [ -n "$MACADDR" ] && sed -i -e "s/MACADDR=.*/MACADDR=$MACADDR/g" \
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  [ -n "$HWADDR" ] && sed -i -e "s/HWADDR=.*/HWADDR=$HWADDR/g" \
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  # Regenerate the SSH keys on the new clone's root filesystem
  [ -n "$VERBOSE" ] && echo "Regenerating SSH keys in $CLONE_MNT_PT/etc/ssh/ ..."
  rm -f $CLONE_MNT_PT/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_key

  copy_key
 }
```

```
#+--------------------------------------------------------------------------+
function set_online
# This will set online the target minidisk.
#   Arg1 - Minidisk virtual address to set online
#+--------------------------------------------------------------------------+
 {
  local target_mdisk=$(echo $1 | tr 'A-Z' 'a-z')
  chccwdev -e 0.0.$target_mdisk >/dev/null
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: chccwdev -e 0.0.$target_mdisk failed with $rc - exiting"
    abort 1
  fi

  local target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" | awk
'{ print $7 }'`
  if [ "$target_dev_node" = "" ]; then
    echo "Error: can't find $target_mdisk(ECKD) in /proc/dasd/devices - exiting"
    set_offline $target_mdisk
    abort 1
  fi
 }

#+--------------------------------------------------------------------------+
function set_offline
# This will set offline the target minidisk.
#   Arg1 - Minidisk virtual address to set offline
#+--------------------------------------------------------------------------+
 {
   target_mdisk=$(echo $1 | tr 'A-Z' 'a-z')
   chccwdev -d 0.0.$target_mdisk > /dev/null 2>&1
   rc=$?
   #if [ $rc -ne 0 ]; then
   # echo "Error: chccwdev -d 0.0.$1 failed with $rc - ignoring"
   #fi

   return $rc
 }

#+--------------------------------------------------------------------------+
function mount_cloned_image
# This will mount the cloned root filesystem.  It will pair a minidisk
# address to a device file and then mount the first partition.
#   Arg1: The minidisk address to mount
#+--------------------------------------------------------------------------+
 {
  target_mdisk=$1

  target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" | awk '{
print $7 }'`

  wait_for_device /dev/${target_dev_node}1
  [ $? -ne 0 ] && echo "Error: timed out waiting for /dev/${target_dev_node}1" &&
abort 1
```

```
  /bin/mount /dev/${target_dev_node}1 $CLONE_MNT_PT
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

  /bin/mount | grep /dev/${target_dev_node}1 >/dev/null 2>&1
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

 }

#+--------------------------------------------------------------------------+
function mount_cloned_image_lvm
# This will mount the cloned root filesystem.  It will pair a minidisk
# address to a device file and then mount the first partition.
#   Arg1: The minidisk address to mount
#+--------------------------------------------------------------------------+
 {
  target_mdisk=$1

  /bin/mount /dev/$VG_NAME/$LV_ROOT $CLONE_MNT_PT
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

  /bin/mount | grep $LV_ROOT >/dev/null 2>&1
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

 }

#+--------------------------------------------------------------------------+
function umount_cloned_image
#   Unmount the cloned root filesystem
#+--------------------------------------------------------------------------+
 {
  /bin/umount $CLONE_MNT_PT >/dev/null 2>&1

  return $?
 }

#+--------------------------------------------------------------------------+
function check_for_conf
# Check that the configuration file exists for the ID that we are cloning to.
#+--------------------------------------------------------------------------+
 {
  if [ ! -f /etc/clone/${target_linux_id}.conf -a "$PROMPT" != "y" ]; then
   echo "Error: /etc/clone/${target_linux_id}.conf not found. Exiting"
   exit
  fi
 }

#+--------------------------------------------------------------------------+
function check_for_vmcp
# Check that the vmcp module is loaded and the vmcp binary is installed.
#+--------------------------------------------------------------------------+
 {
   # Check that vmcp exists and is executable
   [ ! -x /sbin/vmcp ] && echo "Error: can't find /sbin/vmcp" && exit

   # Load the vmcp kernel module if not already loaded
```

```
   if ! /sbin/lsmod | grep vmcp > /dev/null 2>&1 ; then
     if ! /sbin/modprobe vmcp > /dev/null 2>&1 ; then
       echo "Error: unable to load module vmcp, check kernel version"
       exit
     fi
   fi

  wait_for_device /dev/vmcp
  [ $? -ne 0 ] && echo "Error: timed out waiting for /dev/vmcp" && exit
 }

#+--------------------------------------------------------------------------+
function wait_for_device
# Sleep until a certain file exists
#   Arg1: The path of the file to sleep on.
#+--------------------------------------------------------------------------+
 {
  device=$1

  sleep 2
  for t in $(seq 1 20)
  do
    [ -e $device ] && return 0
    sleep 1
  done
  return 1
}

#+--------------------------------------------------------------------------+
function autolog
# Issue an XAUTOLOG command to bring up the new cloned image.
#+--------------------------------------------------------------------------+
 {
  cp_cmd XAUTOLOG $target_linux_id
  rc=$?
  if [ $? != 0 ]; then
    echo "xautolog $target_linux_id failed with $rc"
    return 0
  fi
  echo "Booting $target_linux_id"
 }

#+--------------------------------------------------------------------------+
# main()

# Only root can run this script
[ $(id -u) != "0" ] && echo "Error: you must be root" && exit

# Check if the user has defined any clone.sh configurations
[ -f /etc/sysconfig/clone ] && . /etc/sysconfig/clone

# Set defaults for clone.sh configurations
[ -z "$PROMPT" ] && PROMPT="y"
[ -z "$CLONE_MNT_PT" ] && CLONE_MNT_PT="/mnt/clone"
```

```
# If the clone mount point does not exist then we'll create it for you
[ ! -d $CLONE_MNT_PT ] && mkdir -p $CLONE_MNT_PT

# Check if -v was specified on the command line
if [ "$1" = "-v" ] ; then
  VERBOSE=1
  shift
fi

# If no command line options were provided show the help message
[ $# -eq 0 ] && help

# If one comnand line option was provided show the help message
if [ $# -lt 2 ]; then
  echo "Error: incorrect number of arguments"
  help
fi

# Check that vmcp exists and the module is loaded
check_for_vmcp

# Allow UPPER or lower case source, target, blacklist entries.
# Convert all to lower case for consistency.
source_linux_id=$(echo $1 | tr "[:upper:]" "[:lower:]")
target_linux_id=$(echo $2 | tr "[:upper:]" "[:lower:]")

# Check the blacklist, which prevents using the master image as a target.
if [ -f /etc/clone/blacklist.conf ]; then
  . /etc/clone/blacklist.conf
  BlackList=$(echo ${BLACKLIST} | tr "[:upper:]" "[:lower:]")
  for Target in ${BlackList}
  do
    if [ "${Target}" == "${target_linux_id}" ]; then
      echo "${target_linux_id} is blacklisted!  Exiting!"
      exit
    fi
  done
fi

# Check that the master and clone z/VM IDs are logged off.
check_logged_off $source_linux_id
check_logged_off $target_linux_id

# Check that the clone's configuration file exists
check_for_conf

# Collect information from the clone's configuration file
get_target_info $@
[ "$PROMPT" = "y" ] && ask_are_you_sure

echo "Cloning $source_linux_id to $target_linux_id ..."
[ -z "$DASD" ] && echo "Error: no DASD defined in
/etc/clone/${target_linux_id}.conf" && exit
copy_disks $DASD
```

```
# Update the newly cloned image locally, so link, set online then mount the
# clone's root filesystem.  Then call modify_cloned_image to update
# configuration files with the proper settings.  Finally unmount,
# set offline and unlink the disk.
echo "Updating cloned image ..."
if [ -n "$VG_NAME" ]; then
  link_disks $DASD
  # FIXME wait for disks
  sleep 2
  /sbin/vgscan
  # FIXME wait for vgscan
  sleep 2
  /sbin/vgchange -a y $VG_NAME
  mount_cloned_image_lvm $CLONE_LINK
else
  link_one $target_linux_id $DASD_ROOT $CLONE_LINK W
  set_online $CLONE_LINK
  mount_cloned_image $CLONE_LINK
fi
modify_cloned_image
umount_cloned_image
if [ -n "$VG_NAME" ]; then
  /sbin/vgchange -a n $VG_NAME
  unlink_disks $DASD
else
  set_offline $CLONE_LINK
  unlink_one $CLONE_LINK
fi

# Autolog the clone unless AUTOLOG has been set to "n"
[ "$AUTOLOG" = "y" ] && autolog

echo "Successfully cloned $source_linux_id to $target_linux_id"
```

SLES clone.sh script

This section lists the code for the `/usr/local/sbin/clone.sh` script that clones from a SLES golden Linux image to a target virtual machine.

```
#!/bin/sh
#
# clone.sh <LinuxUserID> - clone a Linux server running under z/VM
#
# For details on how this script works see the book:
#  "z/VM and Linux on IBM System z: The Cloud Computing Cookbook
#    for z/VM 6.3 RHEL 6.2 and SLES 11 SP3"
#  on the Web at: http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.pdf
#
# -----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
```

```
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# -----------------------------------------------------------------------------

#+--------------------------------------------------------------------------+
function help()
# give help
#+--------------------------------------------------------------------------+
 {
  echo "Usage: clone [options] from <sourceID> to <targetID>"
  echo ""
  echo "  Clone Linux from sourceID 100 and 101 minidisks to targetID"
  echo "  options:"
  echo "    -v or --verbose: verbose"
  echo ""
  echo "Example: clone.sh from s11gold to linux01"
  exit 1
 }

#+--------------------------------------------------------------------------+
function processArguments()
# Parse command line arguments
# Args: The arguments passed in to the script
#+--------------------------------------------------------------------------+
 {
  verbose="off"
  sourceID="none"
  targetID="none"
  while (( "$#" )); do
    case $1 in
      -v|--verbose)
         verbose="on"
         ;;
      from)
        shift
        sourceID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold source ID to upper case
        ;;
      to)
        shift
        targetID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold target ID to upper case
        ;;
    esac
    shift
  done
  if [ $sourceID = "none" ]; then # source user ID was not passed
    echo "Error: Source Linux user ID not supplied"
    help
  fi
  if [ $targetID = "none" ]; then # target user ID was not passed
    echo "Error: Target Linux user ID not supplied"
    help
  fi
```

```
 }

#+--------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
  echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print
$2}'`
  return $retVal
 }

#+--------------------------------------------------------------------------+
function checkID()
# Verify user ID exists and is logged off
#    Arg 1: The user ID to check
#+--------------------------------------------------------------------------+
 {
  userID=$1
  echo "Checking that $userID exists and is not logged on ..."
  CPcmd QUERY $userID
  rc=$?
  case $rc in
    0)  # user ID is logged on or disconnected
      echo "$userID user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$userID user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "Return code of $rc unexpected from QUERY $userID"
      echo "User ID must exist and be logged off"
      exit 4
  esac
 }

#+--------------------------------------------------------------------------+
function prepareIPaddr()
# Set the variable "newIPaddr" by adding a backslash before any "."s
#    Arg 1: The IP address to be modified
#+--------------------------------------------------------------------------+
 {
  newIPaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }
```

```
#+--------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+--------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function copyDisk()
# Use FLASHCOPY to copy a disk, if it fails, fall back to dasdfmt then dd
#   Arg 1: Source vaddr
#   Arg 2: Target vaddr
#+--------------------------------------------------------------------------+
 {
  source=$1
  target=$2
  echo ""
  echo "FLASHCOPYing $source to $target ..."
  CPcmd FLASHCOPY $source 0 end to $target 0 end
  if [ $? != 0 ]; then
    echo "FLASHCOPY failed, falling back to dasdfmt and dd ..."
    chccwdev -e $source
    if [ $? != 0 ]; then exit 7; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 8; fi
    sleep 1
    srcDev=/dev/$(egrep ^0.0.$source /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 5; fi
    tgtDev=/dev/$(egrep ^0.0.$target /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 6; fi
    echo "dasdfmt-ing $tgtDev ..."
    dasdfmt -y -b 4096 -f $tgtDev
    if [ "$?" != 0 ]; then exit 9; fi
    echo "dd-ing $srcDev to $tgtDev ..."
    dd bs=1M if=$srcDev of=$tgtDev oflag=sync
    if [ "$?" != 0 ]; then exit 10; fi
    sync
    echo "disabling and re-enabling $target ..."
    chccwdev -d $target
    if [ $? != 0 ]; then exit 11; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 12; fi
    sync
```

```
  fi
 }

#+--------------------------------------------------------------------------+
function askAreYouSure()
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: Minidisks 100 and 101 will be copied to $targetID"
  echo "Network data is retrieved from $targetID PARM-S11 on 191 disk"
  echo "during the first boot of $targetID"
  echo -n "Are you sure you want to overwrite these disks (y/n): "
  read ans
  if [ $ans != "y" ]; then
    echo "Aborting clone per user input"
    exit 16
  fi
 }

#+--------------------------------------------------------------------------+
function copySystem()
# For each of two minidisks 100 and 101:
#   -) Link disk
#   -) Enable disk
#   -) Copy disk
#+--------------------------------------------------------------------------+
 {
  echo "Linking source and target 100 disks ..."
  CPcmd detach 1100
  CPcmd link $sourceID 100 1100 rr
  if [ $? != 0 ]; then exit 17; fi
  CPcmd detach 2100
  CPcmd link $targetID 100 2100 mr
  if [ $? != 0 ]; then exit 18; fi
  echo "Copying 100 disks ..."
  copyDisk 1100 2100
  echo "Take 1100 Offline...."
  chccwdev -d 1100
  CPcmd det 1100
  CPcmd det 2100

  echo " "
  echo "-------------------------------------"
  echo "Linking source and target 101 disks ..."
  CPcmd detach 1101
  CPcmd link $sourceID 101 1101 rr
  if [ $? != 0 ]; then exit 19; fi
  CPcmd detach 2101
  CPcmd link $targetID 101 2101 mr
  if [ $? != 0 ]; then exit 20; fi
  echo "Copying 101 disks ..."
  copyDisk 1101 2101
  echo "Taking 1101 Offline..."
  chccwdev -d 1101
```

```
  CPcmd det 1101
  echo "Taking 2101 Offline..."
  chccwdev -d 2101
  CPcmd det 2101
 }

# main()
processArguments $@                          # process arguments passed by user
if [ $verbose = "on" ]; then set -vx; fi  # turn on debug
checkID $sourceID                            # user ID must exist and be logged off
checkID $targetID                            # user ID must exist and be logged off
# getNetworkInfo                               # get info from parm files
askAreYouSure                                # confirm disks will be overwritten
copySystem                                   # copy source disks to target
# modifyClone                                  # modify newly copied system
echo "sleeping 10 seconds"
sleep 10
CPcmd XAUTOLOG $targetID                     # bring new clone to life
if [ $verbose = "on" ]; then set +vx; fi  # turn off debug
echo "Successfully cloned $sourceID to $targetID"
exit 0
```

SLES boot.clone script

This section lists the code for the `/etc/init.d/boot.clone` script that runs at "first boot" of a newly cloned SLES system.

```
#!/bin/bash
#
# /etc/init.d/boot.clone
#
### BEGIN INIT INFO
# Provides:          boot.clone
# Required-Start:    boot.localfs boot.rootfsck
# Required-Stop:     boot.localfs
# Default-Start:     B
# Default-Stop:
# Short-Description: Change configuration during boot
# Description:       Change the current configuration of the system
#  during first bootup. This script works as follows:
#  1. Run vmcp q userid
#  2. Search for a cms file called userid() PARM-S11
#  3. Get new values for network config from there
#  4. Update the network configuration accordingly
#  This previously used to be the cloning.sh script on linuxadmin.
### END INIT INFO

. /etc/rc.status

rc_reset

#+--------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
```

```
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
# echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print
$2}'`
  return $retVal
 }

#+--------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+--------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function getUserid()
# Read current userid with vmcp q userid
#+--------------------------------------------------------------------------+
 {
 modprobe vmcp
 UserID=$(CPcmd q userid | awk '{print $1}')
 echo $UserID
 }

#+--------------------------------------------------------------------------+
function getNetworkInfo()
# Bring 191 minidisk online to check for my parameter files
#+--------------------------------------------------------------------------+
 {
  # recycle 191 to pick up latest changes
  chccwdev -d 191
  chccwdev -e 191
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "unable to enable 191, rc from chccwdev = $rc"
    exit 13
  fi
  udevadm settle
```

```
  CMSdisk=`lsdasd | grep 0191 | awk '{ print $3 }'`
  cmsfslst -d /dev/$CMSdisk | grep -i $1 | grep PARM-S11
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $1 PARM-S11 not found on 191 minidisk. Exiting"
    exit 14
  fi

# get information about target
  { while read parameter; do
    #echo "parameter: ${parameter%=*}"
    case "${parameter%=*}" in
      Hostname)
        targetHostname=${parameter#*=}
        ;;
      HostIP)
        targetIP=${parameter#*=}
        ;;
      Nameserver)
        targetDNS=${parameter#*=}
        ;;
      Gateway)
        targetGW=${parameter#*=}
        ;;
      Netmask)
        targetMask=${parameter#*=}
        ;;
      Broadcast)
        targetBroadcast=${parameter#*=}
        ;;
      ReadChannel)
        prepareVaddr ${parameter#*=}
        targetReaddev=$newVaddr
        ;;
      WriteChannel)
        prepareVaddr ${parameter#*=}
        targetWritedev=$newVaddr
        ;;
      DataChannel)
        prepareVaddr ${parameter#*=}
        targetDatadev=$newVaddr
        ;;
      *)
        # don't know about any other parameters
        ;;
    esac
  done < <(cmsfscat -a -d /dev/$CMSdisk $1.PARM-S11 | tr '[:space:]' '\n')
  }
 }

#+--------------------------------------------------------------------------+
function createNetworkConfig()
# - remove existing network configuration if it exists
# - create new network configuration from information in CMS parmfile
# - update HOSTNAME, hosts, and resolv.conf
```

```
#+--------------------------------------------------------------------------+
 {
 # delete old configuration
 rm -f /etc/sysconfig/network/ifcfg-eth0
 # setup new configuration
 if [ -n "${targetHostname}" ]; then
   echo "Setting hostname to ${targetHostname}"
   echo ${targetHostname} > /etc/HOSTNAME
 fi
 if [ -n "${targetDNS}" ]; then
   echo "Setting dns resolver to ${targetDNS}"
   sed -i '/nameserver/d' /etc/resolv.conf
   echo "nameserver ${targetDNS}" >> /etc/resolv.conf
 fi
 # echo target stuff
 # will add configuration of different devices when time permits.
 if [ -n "${targetIP}" ]; then
   echo "Setting IP address to ${targetIP}"
   echo "STARTMODE='onboot'" >> /etc/sysconfig/network/ifcfg-eth0
   echo "BOOTPROTO='static'" >> /etc/sysconfig/network/ifcfg-eth0
   echo "IPADDR='${targetIP}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetMask}" ]; then
   echo "Setting netmask to ${targetMask}"
   echo "NETMASK='${targetMask}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetBroadcast}" ]; then
   echo "Setting broadcast to ${targetBroadcast}"
   echo "BROADCAST='${targetBroadcast}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetGW}" ]; then
   echo "Setting default gateway to ${targetGW}"
   sed -i '/default/d' /etc/sysconfig/network/routes
   echo "default ${targetGW} - -" >> /etc/sysconfig/network/routes
 fi
 }
#+--------------------------------------------------------------------------+
function cleanupSSH()
# - remove all existing ssh keys
#+--------------------------------------------------------------------------+
 {
# Delete SSH keys - sshd will recreate them at first boot
  echo "Removing SSH keys"
  rm /etc/ssh/ssh_host*
 }


case "$1" in
    start)
   # update system configuration
        userid=$(getUserid)
        getNetworkInfo $userid
        createNetworkConfig
        cleanupSSH
        chkconfig boot.clone off
```

```
   rc_reset
   ;;
    stop|restart)
        # this should never happen
        # nothing to do
   ;;
    status)
   # probably never will be run.
        # nothing to do
   ;;
    *)
   echo "Usage: $0 {start}."
   exit 1
   ;;
esac

rc_exit
```

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that most publications that are referenced in this list are available in softcopy only:

- *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux 7.1 Servers*, SG24-8303
- *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354
- *Virtualization Cookbook for IBM Z Volume 5: KVM*, SG24-8463
- *Fibre Channel Protocol for Linux and z/VM on IBM System z, SG24-7266*
- *Security on z/VM, SG24-7471*
- *Running Linux Guest in less than CP Privilege Class G*, REDP-3870
- *Sharing and maintaining Linux under z/VM, REDP-4322*
- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- *Accounting and Monitoring for z/VM Linux guest machines, REDP-3818*
- *Systems Management APIs for z/VM, REDP-3882*
- *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR), SG24-8006*
- *IBM Wave for z/VM Installation, Implementation, and Exploitation, SG24-8192*
- *Introduction to the New Mainframe: z/VM Basics, SG24-7316*
- *z/VM and Linux on IBM System z*, SG24-7492
- *Using z/VM for Test and Development Environments: A Roundup, SG24-7355*
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864
- *Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- *z/VM Performance Toolkit Guide*, SC24-6156
- *IBM z/VM V6R3 Installation Guide,* GC24-6246
- *The Program Directory for Performance Toolkit for VM*, GI10-0785
- *z/VM Performance Toolkit Reference*, SC24-6157
- *z/VM Getting Started with Linux on System z*, SC24-6194
- *IBM z/VM CP Planning and Administration*, SC24-6178
- *z/VM: CMS and REXX/VM Messages and Codes,* GC24-6118
- *z/VM CP Commands and Utilities Reference*, SC24-6175
- *z/VM TCP/IP Planning and Customization*, SC24-6125
- *Environmental Record Editing and Printing Program (EREP): Reference*, GC35-0152
- *Environmental Record Editing and Printing Program (EREP): User's Guide*, GC35-0151
- *Getting Started With Linux on System z*, SC24-6096
- *z/VM Security and Integrity* paper:

  http://ibm.com/vm/library/zvmsecint.pdf
- *z/VM Guide for Automated Installation and Service,* GC24-6197
- *z/VM Service Guide*, GC24-6232
- *z/VM RACF Security Server Auditor's Guide,* SC24-6212:

  http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSR8C10.pdf

Online resources

The following websites are also relevant as further information sources:

- z/VM:
  - Installation:

    http://www.ibm.com/vm/install
  - Publications:

    http://www.ibm.com/vm/pubs
  - Technical library:

    http://www.ibm.com/vm/library
  - Security:

    http://www.ibm.com/vm/security
  - Performance:

    http://www.ibm.com/vm/perf
  - Performance tips:

    http://www.ibm.com/vm/perf/tips

- Peer collaboration and shared community knowledge
  - Linux for z Systems: The *Linux/390 project* website and associated wiki:
    - http://linuxvm.org/
    - http://wiki.linuxvm.org/
  - List servers; including `IBMVM`, `Linux-390`, `VM-UTILS`, and more:

    http://www.ibm.com/vm/techinfo/listserv.html
- IBM Techdocs technical sales library:

  http://www.ibm.com/support/techdocs
- Linux distributions:
  - Red Hat:
    - Documentation for z Systems Linux Development stream:

      http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html
    - General information:

      http://www.redhat.com/en/resources/red-hat-enterprise-linux-ibm-system-z
    - No-charge evaluation download for IBM z Systems:

      http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux
  - SUSE:
    - Documentation for z Systems Linux Development stream:

      http://ibm.com/developerworks/linux/linux390/documentation_suse.html
    - General information:

      http://www.suse.com/products/systemz
    - SLES on IBM z Systems forum:

      http://forums.suse.com/forumdisplay.php?42-SLES-for-System-Z
    - No-charge evaluation download for IBM z Systems:

      http://www.suse.com/products/server/download/
  - Ubuntu:
    - Documentation for z Systems Linux Development stream:

      http://ibm.com/developerworks/linux/linux390/documentation_ubuntu.html
    - Ubuntu Linux Server Guide (LTS releases only):

      http://help.ubuntu.com/lts/serverguide

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Services

ibm.com/services

IBM wants your input

Do you have a suggestion on how z/VM or a related product could be made even better? Is there a specific feature you would like to have? Would you like the opportunity to collaborate directly with the IBM product development teams and other product users? The RFE Community is the place.

What is the RFE Community?

The RFE community is a place where you can collaborate directly with product management teams and other product users through your ability to search, view, comment on, submit, and track product Requests For Enhancement (RFEs).

How can the RFE community help you?

Using the RFE Community as the method for enhancement submission provides the following improvements:

- Eliminates multiple touch points that slow down communication in the enhancement process.
- Increases the transparency in the development process for you
- Provides predictable response times for enhancement requests
- Empowers you to influence product direction and road maps and improve communication between users and development
- Provides a better tool for you to review other product enhancement ideas and cast a vote for your favorites
- Avoids the need to open a Problem Management Report (PMR) for a simple enhancement request
- Avoids the need to call Customer Support to determine the status of a specific RFE
- Provides the ability to comment on, and provide workarounds for, RFEs that are created by other customers
- Provides the ability to see comments on RFEs from other customers

What are some of the popular features of the RFE community?

- Online Submission of RFEs
- Browse RFEs by product
- Top 20 Watched RFEs
- Top 20 Voted RFEs
- Planned RFEs
- Delivered RFEs
- Online Searching for RFEs
- Vote on RFEs
- Watch RFEs
- Set Email or RSS feed notifications
- Comment on RFEs
- Start or join a group to discuss RFEs

Join today

Visit the RFE Community on the DeveloperWorks section of ibm.com:

http://ibm.com/developerworks/rfe

Redbooks

Virtualization Cookbook: Vol 1: IBM z/VM 7.2

SG24-8147-02

ISBN 0738459720

IBM

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

SG24-8147-02

ISBN 0738459720

Printed in U.S.A.