

The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3

Lydia Parziale

Berthold Gunreben

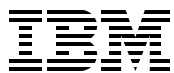
Filipe Miranda

Paul W Novak

Ken Werner



z Systems



International Technical Support Organization

**The Virtualization Cookbook for IBM z Systems
Volume 1: IBM z/VM 6.3**

August 2015

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

Second Edition (August 2015)

This edition applies to Version 6, Release 3 of IBM z/VM, Red Hat Enterprise Linux version 7.1, and SUSE Linux Enterprise Server 12.

© Copyright International Business Machines Corporation 2011, 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
IBM Redbooks promotions	xi
Preface	xiii
Concept of the series	xiii
Volumes in this series	xiii
Volume 1 main chapters:	xiv
Conventions	xiv
Font conventions that are used in this book	xiv
Command conventions that are used in this book	xv
Operating system releases that are used	xv
Authors	xv
Special thanks	xvi
Now you can become a published author, too!	xvii
Comments welcome	xvii
Stay connected to IBM Redbooks	xvii
Summary of changes	xix
Summary of changes in this book	xix
Part 1. Using IBM z/VM 6.3	1
Chapter 1. Introduction to Linux on the IBM mainframe under z/VM	3
1.1 What is virtualization?	4
1.2 Why the mainframe, and why z/VM?	4
1.3 The philosophy that was adopted in authoring this book	5
1.4 A high-level overview of components and terminology	5
1.5 z/VM components, capabilities, and enhancements	7
1.6 Choices and decisions for this book	10
1.7 Single system image design	10
1.8 Infrastructure design	11
1.9 Usability tests that are performed	13
1.10 Understanding LOGOFF versus DISCONNECT	13
1.11 Summary of Linux and z/VM similarities	13
Chapter 2. Planning	15
2.1 Planning for VMSSI with LGR	16
2.1.1 Hints and tips	16
2.1.2 The need for ECKD DASD	18
2.2 Bill of materials	18
2.2.1 Hardware resources	18
2.2.2 Software resources	19
2.2.3 Networking resources	19
2.3 z/VM standardized conventions	20
2.3.1 DASD volume labeling convention	20
2.3.2 Backup file naming convention	21
2.3.3 Command retrieve convention	22

2.4	Disk planning	22
2.5	HiperDispatch planning	24
2.6	Storage (memory) planning	24
2.6.1	Considerations for z/VM 6.3 initial installation and migrations	24
2.6.2	Storage allocation	25
2.6.3	Global aging list	26
2.7	Paging planning	26
2.7.1	Recommendations, tips, and hints	26
2.7.2	Calculating paging space	27
2.8	Password planning	28
2.9	Network planning	29
2.10	Channel-to-channel adapter planning	30
2.11	Architectural overview of the environment that is used in this book	31
2.12	Planning worksheet	31
2.12.1	IBM Shopz	32
2.12.2	Hardware Management Console	32
2.12.3	z/VM Installation Planning Panels (INSTPLAN)	32
2.12.4	z/VM networking resources	35
2.12.5	z/VM DASD	36
2.12.6	FCP devices	36
2.12.7	Linux resources	38
2.12.8	Host names and IP addresses	38
Chapter 3.	Configuring a workstation for mainframe access	41
3.1	PuTTY: A no-charge SSH client for Microsoft Windows	42
3.2	Set up a VNC client	48
3.3	3270 emulators	49
3.4	Linux desktops	50
Chapter 4.	Installing and configuring z/VM	51
4.1	Obtaining z/VM through electronic download	52
4.2	Configuring an FTP server for z/VM installation	54
4.2.1	Create directories on the FTP server and upload the installation image	54
4.3	Installing z/VM from a DVD or an FTP server	55
4.3.1	Start the z/VM installation	56
4.3.2	Copy a plain z/VM system to DASD	59
4.3.3	IPL the first SSI member	65
4.3.4	IPL for the remaining SSI members	66
4.3.5	Initialize the allocated DASD for z/VM Service data	67
4.3.6	Service-level validation and subscribing to service notifications	69
4.4	Configuring the XEDIT PROFILE	70
4.5	Editing the z/VM SYSTEM CONFIG file	74
4.5.1	Modify features and optimize parameter settings	74
4.5.2	Enable and configure virtual networking components	76
4.5.3	Use CPSYNTAX to validate the modified system configuration file	79
4.6	Initial TCP/IP configuration	80
4.6.1	Using the z/VM IPWIZARD tool	80
4.6.2	Configure TCPIP to automatically start during the system IPL	82
4.6.3	Copy the utilities and REXX EXECs to Shared File System (SFS) pools	83
4.7	Adding additional page volumes and perm (user) volumes	86
4.7.1	Format volumes for page space	86
4.7.2	Use the CPFORMAT EXEC	87
4.7.3	Format DASD for minidisks	88

4.7.4	Update the SYSTEM CONFIG file	89
4.7.5	Attach minidisk volumes to the system for use	91
4.8	Enabling the IBM Directory Maintenance Facility (DirMaint)	92
4.8.1	Enable DirMaint	93
4.8.2	Change default passwords	94
4.8.3	Customize the EXTENT CONTROL file	97
4.8.4	Start DirMaint	100
4.8.5	Validation of DirMaint	101
4.9	Implementing additional network features.	103
4.9.1	Enable z/VM FTP and Network File System functionality.	103
4.9.2	Reconfigure TCP/IP for high availability by using a VSWITCH.	104
4.10	Shut down and IPL the SSI cluster again	106
4.10.1	IPL the other SSI members.	108
4.11	Validating and testing your changes	108
4.12	Enabling basic system automation	109
4.12.1	Configure AUTOLOG1's PROFILE EXEC	109
4.12.2	Configuring and enabling the programmable operator facility (PROP).	110
4.12.3	z/VM User Directory PROFILES	115
4.12.4	Creation and use of z/VM User Directory prototypes (PROTODIRs)	118
4.12.5	Create a time-based virtual service machine named CRONSVM.	123
4.12.6	Create VMLOGS.	126
4.12.7	Shut down and IPL the SSI cluster again	127
4.13	z/VM security and hardening.	128
4.13.1	Use an External Security Manager for correct resource security	128
4.13.2	Using LOGONBY for correct accountability	129
4.13.3	Encrypting communications to and from z/VM	132
4.13.4	High-level z/VM security	145
4.13.5	Linux virtual machine privilege classes.	145
4.14	Back up and restore your z/VM system	145
4.15	Create an SFS file pool for Linux virtual machines	146
4.15.1	Characteristics of SFS file pools	146
4.15.2	Add a directory entry for the new SFS server machine.	146
4.15.3	Generate the SFS file pool for Linux guest systems.	148
4.15.4	Add a directory entry for the SFS administration machine	151
4.15.5	Enroll the Linux virtual machines as USERS	152
4.15.6	Add Linux parm files and REXX EXECs to the LNX file pool	153
4.16	Creating identity LNXADMIN for Linux administration.	155
4.17	Monitoring file pool utilization	157
Chapter 5.	Servicing z/VM	159
5.1	Typical release schedule.	160
5.2	Recommended service upgrades (RSUs).	160
5.3	How to apply a recommended service upgrade	162
5.3.1	Get service from the Internet	162
5.3.2	Download the service files	163
5.3.3	Receive, apply, and build the service	164
5.3.4	Put the service into production	166
5.4	How to apply a program temporary fix	168
5.4.1	Get service by using Shopz	168
5.4.2	Determine whether a PTF was applied.	169
5.4.3	Download the service to z/VM.	170
5.4.4	Receive, apply, and build the service	172
5.4.5	Put the service into production	173

5.4.6 Check for APARMEMO files	174
5.5 How to determine the service level of TCP/IP	175
5.6 Moving on to Linux	176
Chapter 6. Planning and preparing for Linux workloads	177
6.1 Planning a Linux virtual machine	178
6.2 Considerations for disk storage types	178
6.2.1 Direct-attached storage devices (DASD)	178
6.2.2 Direct-attached Fibre Channel	180
6.2.3 Emulated DASD	183
6.2.4 Minidisks	188
6.2.5 HyperPAV	190
6.3 Network attachment options and considerations	193
6.3.1 VSWITCH interfaces	193
6.3.2 Direct-attached Open Systems Adapter	194
6.3.3 IBM HiperSockets (HIPERS)	195
6.4 Common DirMaint tasks	196
6.4.1 Characteristics of DirMaint and the user directory	196
6.4.2 Checking the status of DirMaint and subcomponents	196
6.4.3 Add a USER to z/VM by using a prototype	197
6.4.4 Add a USER to z/VM without the use of a prototype	198
6.4.5 Add an IDENTITY to z/VM by using a prototype	198
6.4.6 Add an IDENTITY to z/VM without using prototypes	198
6.4.7 Change the amount of memory that is assigned to a user	199
6.4.8 Modify a user	199
6.4.9 Delete a user	200
6.4.10 Add a minidisk to a user or identity	200
6.4.11 Get a copy of the user directory	200
6.4.12 Get and update the EXTENT CONTROL file	200
6.4.13 Clean up the work units	201
6.4.14 Check the disk map	201
6.4.15 Dedicate crypto domains	202
Part 2. Other topics	203
Chapter 7. z/VM live guest relocation	205
7.1 LGR considerations	206
7.1.1 General considerations before relocation	206
7.1.2 Mandatory memory checking that is performed during relocation	206
7.1.3 Optional memory checking that is performed during relocation	207
7.1.4 Minimizing link and resource contention	207
7.2 Relocate a Linux system	207
Chapter 8. z/VM Systems Management API (SMAPI) and Resource Access Control Facility for z/VM (RACF/VM)	209
8.1 Configure DirMaint	210
8.1.1 Set up basic SMAPI configuration	212
8.1.2 Disable support for ensembles	213
8.1.3 Start SMAPI at IPL time	214
8.1.4 Test SMAPI from the Conversational Monitor System (CMS)	215
8.1.5 Test SMAPI from Linux by using smacient	216
8.2 Enable and configure RACF	217
8.2.1 Create the RACF command file	218
8.2.2 Customize SMF	220

8.2.3	Delete the ICHRCX02 exit	222
8.2.4	Copy the RACF databases	224
8.2.5	Set up the AUTOLOG1 and AUTOLOG2 virtual machines.	229
8.2.6	Enable RACF	230
8.2.7	Put RACF into production on all members	231
8.2.8	Configure SMAPI to work with RACF	238
8.3	Verify that DirMaint and RACF work together	241
8.3.1	Configure LogonBy processing	241
8.3.2	Use the RACF SMF data unload utility	244
Chapter 9.	Monitoring z/VM and Linux	245
9.1	Use basic z/VM commands	246
9.1.1	Use the INDICATE command	246
9.1.2	Use other basic commands	248
9.2	z/VM Performance Toolkit	249
9.2.1	Configure the IBM Performance Toolkit for VM	250
9.2.2	Configure web browser support	251
9.2.3	Configure PERFSVM	252
9.2.4	Start the IBM Performance Toolkit for VM	254
9.2.5	Use the IBM Performance Toolkit for VM	255
9.3	Collect and use raw CP monitor data	257
9.3.1	Collect CP monitor data	258
9.3.2	Use CP monitor data	260
9.4	Monitor Linux performance for troubleshooting	262
9.4.1	Monitor Linux performance from z/VM	263
9.4.2	Monitor Linux performance from inside Linux	264
Chapter 10.	Working with disks	269
10.1	Add disk space to virtual machines	270
10.1.1	Make new minidisks or count key data DASD available in Linux	270
10.1.2	Make new emulated DASD (EDEV) available in Linux	272
10.1.3	Make new zFCP LUN available in Linux	273
10.2	Add a logical volume	276
10.2.1	Create a logical volume and file system	276
10.2.2	Update the file system table	279
10.3	Extend an existing logical volume	280
10.4	Moving a physical volume	283
Chapter 11.	Working with networks	287
11.1	Adding CTCAs to an SSI cluster	288
11.1.1	Add the CTC devices dynamically	289
11.1.2	Add the CTC devices permanently	290
11.2	Setting up a private interconnect	291
11.3	Creating a HiperSockets device between logical partitions	291
11.3.1	Verify HiperSockets hardware definitions	291
11.3.2	Create a TCP/IP stack on z/OS	292
11.3.3	Configure the HiperSockets interface on Linux	293
11.3.4	Verify connectivity	295
11.4	Configuring a port group with Link Aggregation Control Protocol	295
11.5	Network commands	298
Chapter 12.	Miscellaneous helpful information	299
12.1	Install a package from the IBM VM Download Library	300
12.1.1	Use the CMS web browser	300

12.2 Manually formatting DASD for use	301
Part 3. Appendixes	305
Appendix A. References, cheat sheets, and blank worksheets	307
Related books	308
Online resources	309
Important z/VM files	309
Cheat sheets	310
XEDIT cheat sheet	310
A vi cheat sheet	312
DirMaint cheat sheet	312
Blank planning worksheet	313
IBM Shopz	313
Hardware Management Console	314
z/VM Installation Planning Panels (INSTPLAN)	314
z/VM networking resources	317
z/VM DASD worksheet	317
Linux resources worksheet	318
Host names and IP addresses worksheet	319
Appendix B. Additional material	321
Locating the web material	322
Using the web material	322
System requirements for downloading the web material	322
Downloading and extracting the web material	323
z/VM REXX EXECs and XEDIT macros	323
CPFORMAT EXEC	323
SSICMD EXEC	328
PROFILE EXEC for Linux virtual machines	330
RHEL64 EXEC	330
SLES11S3 EXEC	331
SWAPGEN EXEC	331
Sample files	331
SAMPLE CONF-RH6 file	331
SAMPLE PARM-RH6 file	331
SAMPLE PARM-S11 file	332
Linux code	332
RHEL clone script	332
SLES clone.sh script	344
SLES boot.clone script	349
Related publications	355
IBM Redbooks	355
Other publications	355
Online resources	356
Help from IBM	357
IBM wants your input!	358

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®	MVS™	VTAM®
DirMaint™	OMEGAMON®	WebSphere®
Domino®	Parallel Sysplex®	XIV®
DS8000®	Power Systems™	z Systems™
ECKD™	PR/SM™	z/Architecture®
eServer™	RACF®	z/OS®
FICON®	Redbooks®	z/VM®
FlashCopy®	Redbooks (logo)  ®	z/VSE®
HiperSockets™	S/390®	z10™
IBM®	System Storage®	z13™
IBM z™	System z®	zEnterprise®
IBM z Systems™	System z10®	
IBM z13™	Tivoli®	

The following terms are trademarks of other companies:

Inc., and Inc. device are trademarks or registered trademarks of Kenexa, an IBM Company.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

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

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

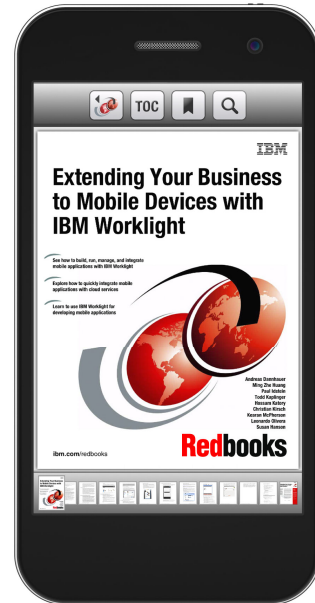
- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

iOS



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication is **volume one of four** in a series of books entitled *The Virtualization Cookbook for IBM z Systems*. The other volumes in the series are:

- ▶ *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux Server 7.1*, SG24-8303
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- ▶ **SG24-8354-00**

It is recommended that you start with Volume 1 of this series as the IBM z/VM® hipervisor is the foundation, or base “layer”, for installing Linux on z Systems. Further details on each volume are presented below.

Concept of the series

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

Volume 1 starts with an introduction, discusses planning, then describes z/VM clustered installation with the *VM Single System Image* (VMSSI or SSI) feature, configuration, hardening, automation, and servicing. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing, configuring, administering, and maintaining the z/VM cluster. This volume also includes a chapter on monitoring both z/VM as well as the Linux virtual servers being hosted.

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

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* introduces the entire concept of Linux on the mainframe system and describes the z/VM platform, planning of, installation into, and configuration of a two-member SSI with z/VM 6.3.
For Volume 1, you need at least two IBM z Systems logical partitions (LPARs) with associated resources and z/VM 6.3 installation media.

- ▶ *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux Server 7.1* describes the installation and customization of REL.

For Volume 2, you will need the Red Hat Enterprise Linux Server (REL) version 7.1 installation media.

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

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

Volume 1 main chapters:

- ▶ Chapter 1, “Introduction to Linux on the IBM mainframe under z/VM” on page 3
This chapter provides a concise introduction to the concept of using the z/VM platform as an enterprise Linux infrastructure on the IBM mainframe.
- ▶ Chapter 2, “Planning” on page 15
This chapter covers the planning of hardware, software, and networking resources that you need to do before you attempt to install z/VM and Linux.
- ▶ Chapter 3, “Configuring a workstation for mainframe access” on page 41
This chapter addresses the configuration of a workstation that is running either Linux or Windows to access the mainframe.
- ▶ Chapter 4, “Installing and configuring z/VM” on page 51
This chapter describes installing z/VM 6.3 as a two-node VM Single System Image feature (VMSSI) cluster, performing the initial configuration, hardening, and enabling basic system automation.
- ▶ Chapter 5, “Servicing z/VM” on page 159
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 is described. Programming Temporary Fixes (PTFs) and Recommended Service Upgrades (RSUs) are both covered.
- ▶ Chapter 6, “Planning and preparing for Linux workloads” on page 177
This chapter describes the necessary steps to begin your first Linux installation. It describes common tasks that are executed during administration, maintenance, and expansion to accommodate additional workloads.
- ▶ Part 2, “Other topics” on page 203 includes chapters on the following subjects:
 - Implementing live guest relocation (LGR) between SSI members
 - Configuring the Systems Management API (SMAP)
 - 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

Conventions

Several highlighting conventions are used in this book.

Font conventions that are used in this book

Monospace and bold

Commands entered by the user on the command line

<code>monospace</code>	Linux file names, directories, and commands
<code>MONOSPACE CAPITALS</code>	z/VM files, virtual machine and minidisk names, and commands
<i>Monospace bold italics</i>	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 <i>example value</i> with the correct value for your system or enterprise.

Command conventions that are used in this book

- ▶ z/VM commands are prefixed with `===>`
- ▶ z/VM XEDIT subcommands are prefixed with `====>`
- ▶ Linux commands that run as root are prefixed with `#`
- ▶ Linux commands that run as non-root are usually prefixed with `$`

Operating system releases that are used

The following releases of operating systems were used in the writing of this book:

z/VM 6.3	GA code, July 2013
RHEL 7.1	GA code, March 2015
SLES 12	GM code, 2015

Authors

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

Lydia Parziale is a Project Leader for the ITSO 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 Linux on z Systems and database management technologies. Lydia is a Certified IT Specialist with an MBA in Technology Management and has been employed by IBM for over 25 years in various technology areas.

Berthold Gunreben is a Build Service Engineer at SUSE in Germany. He has 14 years of professional experience in Linux and is responsible for the administration of the mainframe system at SUSE. In addition to his expertise with Linux on z Systems, he is also a Mainframe System Specialist who is certified by the European Mainframe Academy: <http://www.mainframe-academy.de>. His areas of expertise include high availability on Linux, Realtime Linux, automatic deployments, storage administration on the IBM DS8000®, virtualization systems with Xen, KVM, and z/VM, and documentation. Berthold has written extensively for many of the SUSE manuals.

Felipe Miranda is the Global Lead for Red Hat Enterprise Linux for IBM z Systems and Power Systems™ for Red Hat, Inc.® His key responsibility is to help shape the overall Linux on z Systems and Power Systems strategy for Red Hat. With more than 13 years of experience in Linux and Open Source technologies, he joined Red Hat, Inc. nine years ago and is based in Southern California, US. He holds two degrees, one in Data Communication from University of California, Los Angeles, and another degree in Computer Science from the University Paulista in Sao Paulo, Brazil.

Paul W Novak is a member of the IBM Washington Systems Center z/VM and Linux zGrowth (formerly ATS) team in Endicott, New York. Paul came from IBM Service Delivery where he served as the IBM Global Account webmaster on a team that was responsible for the implementation of IBM WebSphere® and ICS on Linux under z/VM in the world's largest production IBM web middleware environment. He is a Senior Certified IT Specialist with a BSBA in Management Information Systems. Paul held positions in field services, user support, software development, enterprise hosting, and enterprise architecture. Paul has more than 20 years of Linux and Open Source technology experience.

Ken Werner is a Linux specialist working for the Systems unit in IBM Research & Development, Germany. He has seven years of professional experience with Linux on different platforms and has contributed to various Open Source projects, such as the GNU Debugger, libunwind, and OpenEmbedded. Currently, Ken works with the Linux on z development team on the continuous integration, providing the latest code to developers, testers, and performance evaluators.

Special thanks

Thanks to the following people for their contributions:

IBM ITSO Center Poughkeepsie

Dave Bennin, Rich Conway, and Robert Haimowitz

IBM Endicott

Bruce Hayden, Marci Beach, Tim Greer, Emily Hugenbruch, Brian Hugenbruch, Alan Altmark, Brian Wade, Sue Timashenka, Bill Bitner, Kay Blake, Charlie Bryant, Patty Rando, Tung-Sing Chong, Jon Ruhl, Carol Everitt, Bill VanDuzer, Mike Gentile, Dan FitzGerald

IBM Böblingen

Steffen Maier, Pradeep Parameshwaran, Hendrik Brückner, Dominik Klein. Elisabeth Puritscher, Volker Sameske, Ekaterina Teplova

IBM Gaithersburg

Richard Lewis, Fred Bader

Red Hat, Inc.

Chris Mackowski, Jan Stodola, Dan Horak

SUSE

Mike Friesenegger, Mark Post

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 and Poughkeepsie and 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:

- Authors of the previous IBM Redbooks edition, *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147, last updated 22 February 2011: Lydia Parziale, Marian Gasparovic, Berthold Gunreben, Michael MacIsaac, Filipe Miranda, and Daniel Ruutz

- Authors of the previous IBM Redbooks edition, *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1*, SG24-7931, last updated 22 February 2011: Marian Gasparovic and Michael MacIsaac
- Authors of the previous IBM Redbooks edition, *z/VM and Linux on IBM System z: The Virtualization Cookbook for Red Hat Enterprise Linux 6.0*, SG24-7932, last updated 18 February 2011: Brad Hinson and Michael MacIsaac

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

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- 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:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ 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 SG24-8147-01

for The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3

as created or updated on June 27, 2016.

Summary of changes in this book

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

- ▶ The Red Hat chapters were updated and moved to *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux Server 7.1*, 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.
- ▶ 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.
- ▶ 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.

The following changes were made to the setup of IBM z/VM 6.3:

- ▶ The setup of z/VM covers usage of the IBM z/VM Single System Image (VMSSI) feature with a directory management product.
- ▶ A chapter was added about adding a Shared File System (SFS) file pool instead of a shared 191 disk for each of the Linux guests.
- ▶ All of the setup, planning, and service tasks to z/VM 6.3 were reworked and reordered into a workflow that is much faster to complete.
- ▶ A chapter about common IBM DirMaint™ tasks was added.
- ▶ A section about using emulated DASD (EDEV) was added.
- ▶ Chapter 8, “z/VM Systems Management API (SMAPI) and Resource Access Control Facility for z/VM (RACF/VM)” on page 209 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.

The following topic is included for the first time:

- ▶ Systemd in Linux



Part 1

Using IBM z/VM 6.3

This book is volume 1 of a series of three books and starts with an introduction, describes planning, then explains z/VM installation into a two-node single system image (SSI) cluster, configuration, hardening, automation, and servicing.

Volume 1 consists of the following chapters:

- ▶ Chapter 1, “Introduction to Linux on the IBM mainframe under z/VM” on page 3
This chapter provides a concise introduction to the concept of using the z/VM platform as an enterprise Linux infrastructure on the IBM mainframe.
- ▶ Chapter 2, “Planning” on page 15
This chapter covers the planning of hardware, software, and networking resources that you need to do before you attempt to install z/VM and Linux.
- ▶ Chapter 3, “Configuring a workstation for mainframe access” on page 41
This chapter addresses the configuration of a workstation that is running either Linux or Windows to access the mainframe.
- ▶ Chapter 4, “Installing and configuring z/VM” on page 51
This chapter describes installing z/VM 6.3 as a two-node VM Single System Image feature (VMSSI) cluster, performing the initial configuration, hardening, and enabling basic system automation.
- ▶ Chapter 5, “Servicing z/VM” on page 159
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 is described. Programming Temporary Fixes (PTFs) and Recommended Service Upgrades (RSUs) are both covered.
- ▶ Chapter 6, “Planning and preparing for Linux workloads” on page 177
This chapter describes the necessary steps to begin your first Linux installation. It describes common tasks that are executed during administration, maintenance, and expansion to accommodate additional workloads.



Introduction to Linux on the IBM mainframe under z/VM

“Everything should be made as simple as possible, but not simpler.”

— Albert Einstein

This chapter provides a concise introduction to the concept of using the z/VM platform as an enterprise Linux infrastructure on the IBM mainframe.

Initially, it covers virtualization, benefits, and the philosophy that was used in authoring this book. Next, we cover terminology, then a brief summary of z/VM components, capabilities, and enhancements. Finally, the decisions and assumptions that were made by the authors are reviewed, followed by design and usability information.

1.1 What is virtualization?

Virtualization is the ability for a computer system to share resources so that one physical server machine can act as many *virtual servers*. Unlike other hardware environments, the mainframe operates on virtualized hardware by default, resulting in incredible efficiency.

Virtualization is hot in the IT industry. 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.

1.2 Why the mainframe, and why z/VM?

Today, it can be argued that the mainframe is the most functionally rich, reliable, and efficient virtualization platform. The IBM mainframe, z/VM, and its predecessors were virtualized for over five decades.

Over that time, ongoing optimization and refinement brought about qualities that clearly set them apart from the rest:

- ▶ Incredible I/O throughput capabilities: Unlike other virtualization platforms, z/VM guest systems commonly use real hardware access without an additional software layer.
- ▶ Engineered for availability: The hardware in a mainframe is double, sometimes even triple redundant, with a mean time to failure of over 50 years. If a component problem occurs, often you will not know about it until an IBM service support representative shows up to replace it for you.
- ▶ Superior cost savings at scale: With the superior efficiency of mainframe hardware, the result is an environment that brings tremendous ongoing cost savings from software licensing, reduced power, reduced cooling, and a fraction of the floor space versus distributed platforms.
- ▶ When workloads are evaluated from a total cost of ownership (TCO) perspective especially, workloads that are hosted under z/VM on the IBM mainframe are tough to beat.

When Linux came to the IBM mainframe in 2000, it was a natural fit to run under z/VM. You can run hundreds of Linux virtual machines (servers) in the same logical partition (LPAR) under z/VM, only restricted by the amount of available resources. A multi-LPAR mainframe can run thousands of Linux virtual machines, 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 need test environments that are built and rebuilt rapidly to enable them to efficiently deliver their projects, handling 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 the mainframe 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.
- ▶ 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 only 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.

The global z/VM community is known for sharing lessons learned, tips, tools, and advice among each other. You will find the listserv and discussion forums a welcoming and truly useful environment where people are incredibly helpful.

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 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 words. The setup in this book is considered a well-architected environment by the authors. Implementing this environment will result in a robust virtualization environment that can serve as a base for many upcoming new tasks.

1.4 A high-level overview of components and terminology

We provide an overview of the components and technology:

- ▶ Central processor complex (CPC) or sometimes central electronic complex (CEC)

Processor and CPU can refer to either the complete system box, or to one of the central processors (CPUs) within the system box. Although the meaning might be clear from the context of a discussion, even mainframe professionals must clarify which processor or CPU meaning they are using in a discussion. IBM uses the term *central processor complex* (CPC) to refer to the physical collection of hardware that includes main storage, one or more central processors, timers, and channels. Many system programmers use the term *central electronic complex* (CEC) to refer to “the mainframe box”, but the IBM preferred term is CPC.

Mainframe professionals typically use *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) or Integrated Facility for Linux (IFL)) within the CPC. Figure 1-1 on page 6 shows an IBM z13 CPC, which is the physical mainframe.



Figure 1-1 An IBM z13 CPC

- Processing units (PUs)

Briefly, all of the IBM S/390® or z/Architecture® processors within a CPC are processing units (PUs). When IBM delivers the CPC, the PUs are characterized as CPs (for normal work), Integrated Facility for Linux (IFL), Integrated Coupling Facility (ICF) for IBM Parallel Sysplex® configurations, and so on.

- Logical partitions (LPARs)

From the perspective of system use and management, logical partitions (LPARs) are equivalent to separate mainframes/CPCs. Each LPAR has its own operating system, which can be any mainframe operating system. LPARs can also optionally 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 execute in each LPAR. The administrator can also provide weightings for different LPARs, for example, specifying that LPAR1 will receive twice as much processor time as LPAR2.

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, there is no effect on the other LPARs.

- Real storage/central storage

Both of these terms refer to *main system memory*, which is also frequently just called *storage*. Anyone who has previous experience with mainframes and with mobile and tablet devices already understands this concept. 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 CPCs.

- Direct access storage device (DASD)

IBM disk storage subsystems, such as IBM System Storage® DS8300, IBM System Storage DS8700, and IBM XIV® Storage System, provide emulated IBM machine type 3390 disk drives of varying model types. Common models are 03 (3390-03), 09 (3390-09), 27 (3390-27), and 54 (3390-54). The 3390 DASDs are commonly used on current mainframes, and will be covered in much more detail throughout this book.

- The z/VM platform

Within the mainframe (CPC), 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):

- *Control program (CP)*

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 virtual machine, 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)*

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

CP/CMS and Multics, which inspired UNIX, share a common background:

- *Virtual machine (VM)*

From a z/VM perspective, the following terms all refer to the same thing - a virtual machine definition:

User, ID, Identity, Guest, virtual machine, Virtual Server

A *virtual machine 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 machine (VM) will be the preferred term that is used in this book wherever possible.

For additional details, a description of virtual machine types, and the USER and IDENTITY statements, see *z/VM Getting Started with Linux on System z*, SC24-6194.

1.5 z/VM components, capabilities, and enhancements

Many enhancements and new functionality were added to z/VM over the last three releases of the product. The following brief summary describes what is new or enhanced by release version. The currently supported releases are 6.3, 6.2, and 5.4; only the supported releases are described.

z/VM 6.3

z/VM 6.3, which became generally available in late July 2013, extends the mainframe 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 virtual machines
Quadruples memory scalability while continuing to maintain greater than 90% resource utilization
 - Additional vertical scalability to help reduce LPAR sprawl
Considerably more virtual machines 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 utilization of CPU hardware resources for dispatched work.
- ▶ IBM adopted OpenStack as part of its cloud strategy. In concert, 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, which became generally available in December of 2011, continues to help clients extend business value across the enterprise by integrating applications and data while providing high levels of availability, security, and operational ease. This release implements multisystem 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 gives the appearance of being a single system image (SSI).

With the IBM z/VM Single System Image (VMSSI) feature, a running Linux virtual machine can be relocated nondisruptively 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 virtual machines 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 SSI cluster are managed, serviced, and administered as one system. Resources, including the user directory, minidisks, spool files, and network devices, that are used by both the control program (CP) and virtual machines, will be 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 SSI cluster is able to 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-up. 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 SSI cluster, the IBM z/VM Single System Image (VMSSI) feature must be licensed and enabled. Servicing in an SSI 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, avoiding 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.

z/VM 5.4

z/VM 5.4, which became available in August of 2008, provides major improvements when operating on System z servers with large memory configurations. It improves scalability and can help support increased workloads on IBM System z servers. This release uses the capabilities of the IBM System z10®, including the following benefits:

- ▶ Greater flexibility, with support for the new z/VM-mode LPARs, allowing all System z processor-types (CPs, IFLs, System z Integrated Information Processors (zIIPs), System z Application Assist Processors (zAAPs), and ICFs) to be defined in the same z/VM LPAR for use by various guest operating systems
- ▶ Capability to install Linux on System z from the Hardware Management Console (HMC) that eliminates network setup or a connection between an LPAR and the HMC
- ▶ Enhanced physical connectivity by using all OSA-Express3 ports, helping service the network, and reducing the number of required resources

z/VM 5.4 dynamic memory upgrade support allows real memory to be added to a running z/VM system, avoiding the need to shut down z/VM and its guests, deactivate the LPAR, change its memory allocation, reactivate the LPAR, restart z/VM, and restart its guests. Memory can be added nondisruptively to individual guests that support the dynamic memory reconfiguration architecture.

1.6 Choices and decisions for this book

When we were deciding on installing, maintaining, and provisioning (cloning) Linux virtual machines under z/VM, we made many basic decisions. The following list shows several of the choices and assumptions that were made in this book:

- ▶ Use of a commercial systems management product: Because this book is designed for you to learn the basics, the use of a commercial systems management product is not described.
- ▶ In this book, the authors assume that you are going to use a single system image (SSI) environment for continuous operation during service. To simplify operation, prevent administration errors, and provide synchronization of directory data among SSI 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.
- ▶ Proper 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).
- ▶ 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. Certain 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 machines. A read/write `/usr/` file system on the virtual machines 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. The use of conventional 3390 DASD is however still considered to be the fastest and simplest choice.
- ▶ The practice of cloning is no longer considered a good practice and is deprecated. Changes in both RHEL 7 and SLES12 cause cloning to be risky and uncertain. 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 detail.

1.7 Single system image design

With the introduction of z/VM 6.2 in December 2011, the architecture of Linux solutions on z Systems 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 VM Single System Image (VMSSI or SSI) with live guest relocation (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 SSI name is the highest level identifier.

A block diagram of a four member SSI, 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. Four z/VM systems and four system identifiers are in this cluster. However, only one SSI name is in this cluster. In this book, a two member SSI that is installed onto one CPC is described.

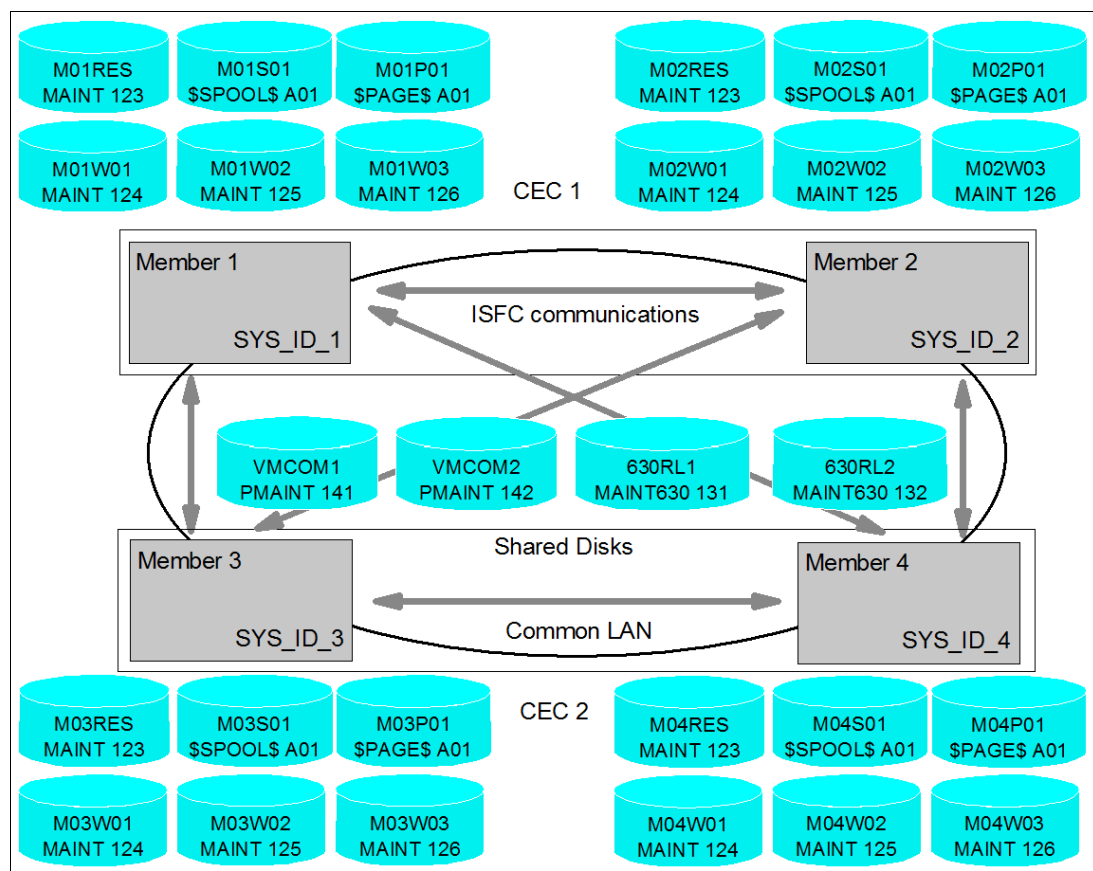


Figure 1-2 z/VM 6.3 single system image block diagram

1.8 Infrastructure design

To install and configure z/VM, and to install, configure, and *provision virtual machines*, a certain infrastructure design must be in place. A z Systems CPC with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-3 on page 12 shows a block diagram of a CPC with z/VM on two LPARs. z/VM comes with many predefined virtual machines. The most important six IDs are shown in the z/VM LPAR above the dashed line. Below the dashed line, you see the virtual machines that are described in this book.

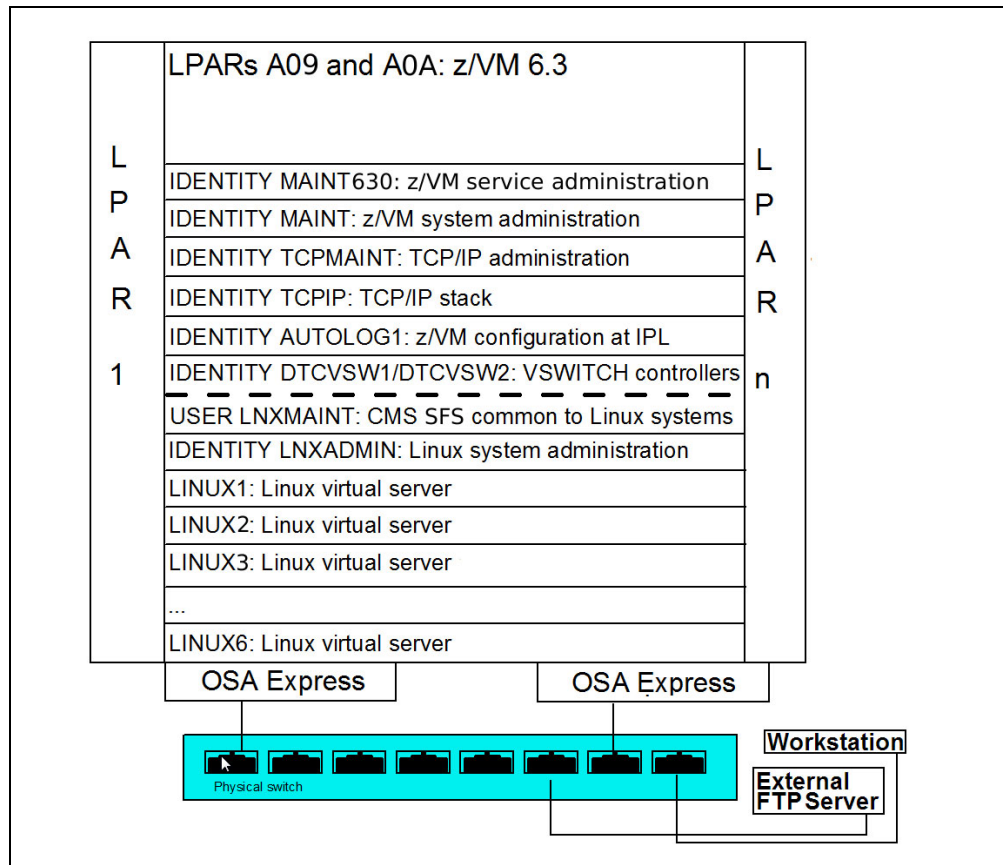


Figure 1-3 System infrastructure and z/VM virtual machines

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

- LNXMAINT** A virtual machine that is used to own and manage the Shared File System (SFS) file pool to be used by Linux virtual machines.
- 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 SSI members at the same time.
- LINUX1 - LINUX4** Four sample *worker* virtual machines.

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

1.9 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 both solid Linux and z/VM system administration skills. By the end of two days, most participants created their first Linux virtual machine. 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.

1.10 Understanding 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. It is analogous to shutting down and powering down a PC when the guest is running CMS. It is analogous to unplugging a running PC if the guest is running Linux.

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

- DO use logoff with system administration virtual machines, such as MAINT630, MAINT, TCPMAINT, and LGLOPR.
- DO NOT use logoff with a VSM providing a service or running automation.
- DO NOT use logoff with a Linux virtual machine while Linux is still running.
- LOGOFF destroys ALL running processes for that virtual machine and terminates any connections.

► DISCONNECT

If you disconnect, your session remains where it is and is resumed when you next execute the login process. It is analogous to turning off a PC monitor, or using a Linux terminal with a utility, such as screen, tmux, or byobu, that permits detachment and reconnection of a running session.

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

1.11 Summary of Linux and z/VM similarities

Although Linux and z/VM differ in many ways, they have similar concepts, functions, and commands, as shown in Table 1-1 on page 14.

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

Linux	z/VM (CP and CMS)
Boot	IPL (initial program load)
Filesystem directory	Disk access mode
Kernel	Control program (CP)
Memory (RAM)	Storage
~/.profile	PROFILE EXEC A
Script (executable file .sh, .ksh, .pl, and similar)	EXEC
Shell	Conversational Monitor System (CMS)
User registry	User directory
vi, vim, emacs, nano, pico, or similar	XEDIT

Table 1-2 shows 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 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)



Planning

“The only reason for time is so that everything doesn’t happen at once.”

— Albert Einstein

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

It begins by describing planning for a *z/VM Single System Image* (VMSSI) cluster followed by discussion of the *bill of materials*, which is a listing of all of the necessary resources. Next, it describes standardized conventions that are adopted for labeling 3390 disk volumes and other 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)

Previous editions of this publication primarily considered z/VM a single stand-alone system with no clustering capabilities. With the increasing popularity of the VMSSI feature, installations of z/VM that do not use VMSSI are declining as time goes on. The authors of this book chose to exclusively cover z/VM with the VMSSI feature enabled.

If you 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 additional nodes to a pre-existing cluster is much easier than having to start from square one.

2.1 Planning for VMSSI with LGR

With z/VM 6.2 and 6.3, Linux virtual servers can be relocated while still running from one z/VM system to another within the same VMSSI cluster. The VMSSI cluster can include up to four logical partitions (LPARs) as members and these LPARs can exist on multiple central electronics complexes (CECs) if you want. This feature to provide mobility is called *live guest relocation* (LGR).

You might need to relocate a running virtual machine 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. Hints are provided to help with the installation of the VMSSI feature, and tips are provided to relocate a Linux guest.

As a preferred practice, define the same real device numbers to reference the same devices on all members of the SSI cluster. You need to discuss this practice with your hardware administrator to ensure that this naming is reflected by the input/output configuration data set (IOCDs).

2.1.1 Hints and tips

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 6.3 with or without the VMSSI feature.

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

To plan and prepare, you are encouraged to use the following IBM publications:

- ▶ Chapter 25 of *IBM z/VM CP Planning and Administration*, Version 6 Release 3, 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 System z*, Version 6 Release 3, SC24-6194

Additionally, consult the z/VM Installation Resources page at this website:

ibm.com/vm/install

An SSI cluster must have direct logical links between all systems. All SSI 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.

Important: We do not cover second-level installation procedures in this cookbook series; however, it is possible for someone more experienced with z/VM to use Volume 1 of this series to do so. For this reason, it is important to be aware of the following:

The z/VM Single System Image feature enables sharing of configuration, parameters, and directory data over Channel-to-channel adapter (CTCA) connections. During the z/VM installation process, when you choose to install a system as second-level, there are customized modifications made to the generated system configuration and user directory parameters. Due to these differences, *you must not create a cluster which contains a mix of first-level and second-level z/VM members. Attempting to do so will result in unpredictable or catastrophic results.* Additional details are covered in *z/VM CP Planning and Administration* (Version 6 Release 3, SC24-6178-05)

Factors that can affect relocation, system performance, or both

Consider the following factors in planning for Linux LGR:

- ▶ **Virtual machine memory:** The size and use of the virtual machine's memory can affect relocation performance. Parts of the processing for relocation are proportional to the size of the virtual machine. The cost of this processing increases with larger virtual machines. Relocation performance is also affected by the frequency and amount of memory that is being changed in the virtual machine.
- ▶ **Matching virtual machine configurations:** To prepare for LGR, ensure that the virtual machine 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 virtual machine's eligibility to relocate, see Chapter 27 of *IBM z/VM CP Planning and Administration*, SC24-6178.
- ▶ **CPU utilization:** The z/VM V6.3 SSI feature will synchronize 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 resource during and after the relocation. You will need to ensure that your paging space is adequate. z/VM 6.3 changed the capabilities and effects of the CP SET RESERVED command and you need to 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 on this aspect of system resources is to issue the CP QUERY ALLOC PAGE command, which will show the percent that is used, the slots that are available, and the slots that are in use. If you add the size of the virtual machines 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 usage can affect system performance negatively. This query command provides only a snapshot in time.
- ▶ **Real memory:** Real memory resources are important for both the source and the destination systems for relocations. You will need enough real memory (1) to hold buffers

during the relocation on both systems, and (2) to accommodate the incoming guest's working set afterward on the target system. Relocation performance will also be affected by the level of overall resource constraint for both the source and destination systems.

- ▶ **Dump space:** If you are allocating a large amount of real storage (memory), you will need to plan for dump space so that if you need to collect a system dump, you will have enough space to write it to disk.
- ▶ **Linux distributions and LGRs:** With the introduction of LGR among members of your SSI, it is increasingly important to identify the level of Linux on z Systems 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.

2.1.2 The need for ECKD DASD

If z/VM 6.3 will be installed into an SSI, 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 an SSI.

For more information about sharing a RACF database, see *z/VM RACF Security Server System Programmer's Guide*, SC24-6212, and for information about DASD sharing, see *IBM z/VM CP Planning and Administration*, SC24-6178.

2.2 Bill of materials

The resources that are needed for a Linux on z Systems project are grouped in the following manner:

- ▶ Hardware
- ▶ Software
- ▶ Networking

2.2.1 Hardware resources

The following hardware is needed:

- ▶ A minimum of one, and up to a maximum of four, z Systems LPARs with:
 - Processors or CPUs per LPAR: One IFL (or CP) minimum; two or more are recommended.
 - Memory: 8 GB storage or more. With z/VM 6.3, expanded storage is no longer recommended, and should not be used. For more information, see the following website:
ibm.com/vm/perf/tips/storconf.html
 - DASD: Three 3390-03s and eighteen 3390-09s were allocated for the reference 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 will act 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 AIX, is recommended. Be aware that a Windows-based FTP server is likely to cause code page translation, resulting in corruption during network transport.

- ▶ A workstation that has network access to the mainframe and operates as the Hardware Management Console (HMC) if physical access is not possible.

2.2.2 Software resources

The following software resources are needed:

- ▶ z/VM 6.3 installation media with documentation. The physical media of DVDs are described. If you will use *Shopz* to download the z/VM installation media and make it available by using an FTP server, physical media will not be needed.
- ▶ Red Hat Enterprise Linux version 7.1 installation media. If you do not have it, you can request a 180-day evaluation copy from Red Hat. See the Appendix , “Online resources” on page 356.
- ▶ SUSE Linux Enterprise Server version 12 installation media (DVD .iso files). If you do not have it, you can request a 180-day evaluation copy at no charge from SUSE. See the Appendix , “Online resources” on page 356.
- ▶ Ubuntu Server version 16.04 LTS installation media. If you do not have it, you can obtain a full copy at no charge from Canonical. See the Appendix , “Online resources” on page 356.
- ▶ The code that is associated with this book, described in full detail inside of Appendix B, “Additional material” on page 321
- ▶ Tools on the workstation:
 - A 3270 emulator such as x3270, Attachmate Extra, Hummingbird Host Explorer, or IBM Personal Communications
 - A Linux Secure Shell (SSH) client, such as PuTTY, xTerm, or similar
 - A Virtual Network Computing (VNC) viewer, such as RealVNC

These resources are described in more detail in the following chapters.

2.2.3 Networking resources

The following network resources are needed:

- ▶ TCP/IP addresses for each z/VM SSI member
- ▶ One TCP/IP address for each Linux virtual machine
- ▶ Associated TCP/IP information:
 - Domain Name Server (DNS) host name
 - DNS domain name
 - DNS server TCP/IP address
 - TCP/IP gateway
 - TCP/IP subnet mask

- TCP/IP maximum transmission unit (MTU) size

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 the management of your network. If no one is responsible for the management of your network, you can use these guidelines for assigning a Locally Administered address range to use with your SSI cluster:

- The address range absolutely 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 your OSA cards will be cabled to. 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, 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.”
- The address must start with “02” in the most significant byte, for example, “020304050607.”
- Do not assign “0000 0000 0000” or “FFFF FFFF FFFF.”
- The range is from 0200 0000 0000 to 02FF FFFF FFFF.

The MAC address range must be unique within the LAN segment you will connect to. For more information, see the IEEE website:

<http://standards.ieee.org/about/get/802/802.3.html>

IMPORTANT: If you plan to install Ubuntu Server using Volume 4 of this series of books, please review Appendix C of for considerations pertaining to firewalls.

2.3 z/VM standardized 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.

2.3.1 DASD volume labeling convention

You need to adopt a standardized convention for labeling DASD. If one or more IBM mainframes are already present, your I/T department might already use a labeling standard that will largely determine the labels to be given to the DASD that is used by your z/VM LPARs.

Each z Systems DASD has a real device address that consists of four hexadecimal digits. Each z Systems DASD has a six character label. It is convenient to include the four-digit address in the label so that you can easily tell the address of each DASD from its label. When followed thoroughly, this convention guarantees that no two DASDs will have the same label,

which can be an important issue, especially when an IBM z/OS® LPAR has access to the DASD.

Sometimes, DASD is shared among LPARs. In which 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 identifies the LPAR or SSI with the first character, which leaves the second character in the label to identify the basic function of the DASD. The example SSI in this book is identified by the character “V”.

The authors of this book are strong advocates of configuring the IODEF in a manner so that each LPAR can see only DASDs that it owns or requires access to.

The 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 in Appendix B, “Additional material” on page 321. This EXEC uses this volume labeling convention. If you want 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'  
...  
*****
```

The 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

2.3.2 Backup file naming convention

It is recommend that you keep copies of important z/VM and Linux configuration files. Always keep copies of original configuration files in case you need to go back to 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, prior to any editing, the original **USER DIRECT** file is copied to the file **USER DIREORIG** before it is modified the first time. The original **SYSTEM CONFIG** file is copied to the **SYSTEM CONFORIG** file. This will ensure you always retain the original copy.

Recent versions

There are two commonly adopted practices for retaining previous versions of files, the “*it works*” method, and, the “*now minus*” method. Both are described below, using the **SYSTEM CONFIG** file as the example, to help you decide which makes the most sense for you:

- ▶ It works
 - Retention of only the most recent working copy, using 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 called just “n minus”
 - Retention of up several recent working copies, using the suffixes -1, -2, -3, and so on.
 - Before editing, SYSTEM CONFIG is copied to the file SYSTEM CONF-1.
 - If SYSTEM CONF-1 already exists, it is renamed to the file SYSTEM CONF-2 first.
 - This is simple to implement using REXX. Example code for NBACKUP EXEC is included

2.3.3 Command retrieve convention

The ability to retrieve past commands is a common tool. Often, it is nice to retrieve in both directions in case you “pass” the command that you are looking for. The default Linux shell, *bash*, does this function by default with 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 virtual machines. No convention exists to retrieve commands in the other direction, but it is possible to set another key to that function. Therefore, F11 is used to *retrieve forward* because it is right next to F12. Also, the same function is useful in the editor, **XEDIT**. The ? subcommand retrieves past commands, so it is recommended that you assign it to F12.

Implementation details for this concept are provided later in this book.

2.4 Disk planning

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)
- ▶ Size of 3390 disks: Mod-3s, Mod-9s, Mod-27s, or larger (such as 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 how 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. Sometimes, a combination of these types of disk storage is used. When a combination is used, the ECKD and EDEV DASD are often used for the root file system and SCSI/FCP disks are used for large data storage areas.

3390-3s, 3390-9s, or larger

Emulated 3390-3s format to about 2.3 GB, and 3390-9s (6.8 GB) and 3390-27s (20 GB) are each three times larger. z/VM 6.3 can be installed on to either 3390-3s or 3390-9s. For PAGE and SPOOL areas, larger disks than 3390-9 can be used; however, all of the involved disks

absolutely must be of identical size and must not serve mixed purposes. Certain larger z Systems client 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).

Disk storage for each Linux image

This version of the book now recommends one 3390-9 DASD that is attached as a full pack minidisk at virtual address 100, which gives about 6.8 GB of space. You might choose to add another full pack 3390-9 DASD at address 101 or 200, which will double the disk space to 13.6 GB.

Additional topics that describe the various options that are available are covered in Chapter 10, “Working with disks” on page 269.

Swapping

The terminology “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. The term 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 has a feature, which is called *virtual disks (VDISKs)*. *VDISKs* exist in memory but are presented to guest operating systems as disks. Mainframe memory is fast (many times faster than disk especially) so using *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 virtual machines must have an adequate set of swap spaces. What defines “an adequate set of swap space” can turn into a lengthy debate. However, the z/VM and Linux community agrees that one or two small swap spaces on virtual disk, which can also 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 clients indicates that correctly sized Linux guests seldom, if ever, will swap. 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 is used. It creates and formats a Linux swap disk from Conversational Monitor System (CMS). Information about the current version, and instructions for downloading and installing **SWAPGEN** are in the `swapgen-readme.txt` file on the Sine Nomine website:

<http://download.sinenomine.net/swapgen>

Notes:

- ▶ A new version of **SWAPGEN** was released by Sine Nomine in the time since the previous edition of this book.
- ▶ Additional considerations about the correct implementation of *VDISK* on Linux systems that use GRUB2 and DRACUT are covered in the Linux installation chapters.

2.5 HiperDispatch planning

In z/VM 6.3, IBM introduced new virtual server dispatching technology, which is called *HiperDispatch*, into z/VM. The z/VM HiperDispatch enhancement is meant to help the workload to get 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.

The overall topic of HiperDispatch is described in great detail at the following page:

<http://www.vm.ibm.com/perf/tips/zvmhd.html>

2.6 Storage (memory) planning

As you proceed through this section, remember the terminology that was covered in 1.4, "A high-level overview of components and terminology" on page 5. *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 z 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 Considerations for z/VM 6.3 initial installation and migrations

Important information about z/VM 6.3 is listed:

- ▶ 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 6.3, this requirement no longer exists.
- ▶ If you are migrating to z/VM 6.3 from an older version, the LPAR activation profile might be configured to define Expanded Storage, or XSTORE. With z/VM 6.3, IBM no longer recommends XSTORE as an auxiliary paging device, and usage of XSTORE with z/VM 6.3 is considered to be deprecated. The aging and filtering functionality that was formerly provided by XSTORE is now provided by z/VM 6.3's Global Aging List (GAL), which is described later in this chapter. If your LPAR has XSTORE defined for it, simply 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, in migrating to z/VM 6.3, you will change to 36 GB of real storage with no XSTORE.

- ▶ z/VM 6.3 changed the capabilities and effects of the **CP SET RESERVED** command, so you will want to review the systems that you are migrating to ensure the values that are in use are still appropriate. Earlier editions of z/VM sometimes failed to honor **CP SET RESERVED** settings for virtual machines, prompting people to oversize the amount of reserved storage that they specified. z/VM 6.3 is more effective and precise in honoring reserved settings.
- ▶ z/VM 6.3 also permits **CP SET RESERVED** for Named Saved Systems (NSSs) or Discontiguous Saved Segments (DCSSs). This new capability was especially intended for the MONDCSS segment. In previous z/VM releases, under heavy storage constraint, MONDCSS was at risk of being paged out and consequently unavailable for catching control program (CP) Monitor records. 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 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 mainframe experience, you might think that the simplest solution is to over-allocate INITIAL storage with the expectation that an overabundance will result 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 will result in needless rework or potential performance problems for you in the future.

An understanding of memory planning is recommended. The following resources cover this important topic:

- ▶ *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926:
<http://www.redbooks.ibm.com/redpieces/abstracts/sg246926.html>
- ▶ The IBM z/VM Performance Resource pages:
<http://www.vm.ibm.com/perf/>
- ▶ The IBM z/VM web page on configuring processor storage:
<http://www.vm.ibm.com/perf/tips/storconf.html>

After you determine the INITIAL value that you will 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 additional storage is typically obtained from *standby storage*, a dynamically calculated value that is based on the following information:

- ▶ Amount of storage that is installed in the CPC that is not currently 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 virtual machine. A good starting point is to set a virtual machine 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 will typically find that Linux under z/VM needs less memory than you are accustomed to.

In addition to that “just enough” amount of memory, assign a number of VDISKS as SWAP disks to each of the Linux virtual machines. The VDISKS need to provide as much memory as Linux will need 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 needs to also be available as real memory to the LPAR, plus page space.

One recommended rule is to have as few virtual machines logged on (or in a disconnected state) as possible to handle the workload that is being presented. Every virtual machine that is not required needs to be logged off where appropriate because more memory is available for the other virtual machines 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 through the **CP SET AGELIST** command or the **STORAGE AGELIST** system configuration file statement.

2.7 Paging planning

Your paging channels and DASD need to 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 in turn becomes increasingly able to handle the concurrent execution of page-fault-inducing virtual machines, resulting in optimal throughput for your workloads.

2.7.1 Recommendations, tips, and hints

- ▶ **No mixing:** A disk volume must be either 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 3390-3s, all your paging volumes must be 3390-3. If you decide to use all 3390-9s, as we did for this book, all of the paging volumes must then be 3390-9s. 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 visibly hurts system performance by restricting the z/VM opportunity for paging I/O concurrency.
- ▶ **Spread out:** If you are planning to deploy workloads that will be frequently paged, spread your paging space over as many 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 will be 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 you are using IBM Fibre Channel connection (FICON®) because this approach can substantially increase throughput.
- ▶ **If Fibre Channel Protocol (FCP) CHPIDs exist and SCSI DASD controllers are installed,** you might consider using 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 comes with 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 you can add additional paging volumes without an IPL, if necessary.

2.7.2 Calculating paging space

The z/VM 6.3 edition of *IBM z/VM CP Planning and Administration*, SC24-6178, was updated to contain a new formula for calculating the amount of paging space to allocate. The formula is shown in Table 2-1.

Table 2-1 Calculations for paging planning

A	Sum of primary address space sizes for all logged-on virtual machines, which are all virtual machines that are typically expected to be up and running under normal conditions, such as DirMaint and other service machines, and the Linux virtual machines.
B	Sum of the sizes of any data spaces for all logged-on virtual machines.
C	Sum of the sizes of any VDISKS that are created for all logged-on virtual machines.
D	Sum of the sizes of any shared NSSes or DCSSes for all logged-on virtual machines.
E subtotal	Add A + B + C + D together 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.

A	Sum of primary address space sizes for all logged-on virtual machines, which are all virtual machines that are typically expected to be up and running under normal conditions, such as DirMaint and other service machines, and the Linux virtual machines.
G	Total number of CP directory pages that are reported by DIRECTXA. Be sure that you convert pages to MB, GB, or whatever unit you are using.
H	10% of real storage that is defined for the LPAR to allow for system-owned virtual pages.
I <i>minimum</i>	Add E + G + H together 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 <i>recommended</i>	Multiply I by a reserve margin value to obtain J , which is the final value you will use. Reserve is added because your calculation might be uncertain and your system will grow 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 Password planning

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

This book considers different system administration roles:

- ▶ z/VM system administrator
- ▶ Linux system administrator
- ▶ Linux virtual machine users

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

The method of backing up z/VM data onto the Linux administration system means that the Linux administrator will have access to all z/VM passwords. Therefore, the examples in this book set all z/VM and Linux system administration passwords to the same value, `linux4vm`, which is a trivial password. You need to use a more complex password in your environment. If the z/VM and Linux system administrator roles must be separate and the Linux administrator cannot access the z/VM passwords, a different method of backing up z/VM data, such as IBM Backup & Restore Manager for z/VM, must be chosen.

You might want to define a finer granularity for passwords that is based on the following system administration roles:

- ▶ The main z/VM system administrator (MAINT, MAINT630, and PMAINT)
- ▶ The z/VM network administrator (TCPMAINT)
- ▶ The Linux virtual machine users (with or without access to 3270 sessions, and with or without the root passwords)

The sets of passwords that you define depend on the roles that your organization adopts and the required governance policies.

2.9 Network planning

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.

Network Administrator

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 network configuration, management, and business controls.

Important: If you have a “network administrator”, you must involve them in your network planning activities.

Collaboration will ensure that MAC addresses and other network configuration will not become a problem that can cause additional reconfiguration later. This should also help to ensure you do not inadvertently create any possible problems.

z/VM Virtual Network Switches (VSWITCH)

Use of VSWITCH is the IBM recommended practice. It is less computationally expensive than the other alternatives and when it is correctly configured, it provides built-in failover. Additionally, it supports 802.1q VLAN by port, 802.1q VLAN by user, port isolation, and 802.3ad link aggregation.

MTU sizes for QDIO Open Systems Adapters (OSAs)

MTU sizing is different from traditional distributed systems. Set MTU 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 **tracpath destination** from a Linux system on the same network segment that your OSAs will use. It traces the path to destination, discovering MTU along this path by using User Datagram Protocol (UDP). It is similar to **traceroute**, but it does not require superuser 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.

Note: These MTU sizes are for both z/VM as well as the Linux virtual servers.

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 and with synchronous operation, SIGA is required for every packet. You will not encounter *packing* (stalling due to queuing) from a VSWITCH as you will from a dedicated OSA.

Additional information that is specific to TCP/IP and the z/VM hypervisor is in Appendix A, “References, cheat sheets, and blank worksheets” on page 307.

Additional information specific to Open Systems Adapters can be found in *IBM z Systems Connectivity Handbook*, SG24-5444. The URL is in Appendix A, “References, cheat sheets, and blank worksheets” on page 307.

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 will 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. The details are covered in 11.1, “Adding CTCAs to an SSI cluster” on page 288.

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

In Figure 2-1, channel paths **4C** and **4D** were provided with attached FCTC control units **47E0** and **57E0** on the **ZVM63A** system, and FCTC control units **4120** and **5120** on **zVM63B**.

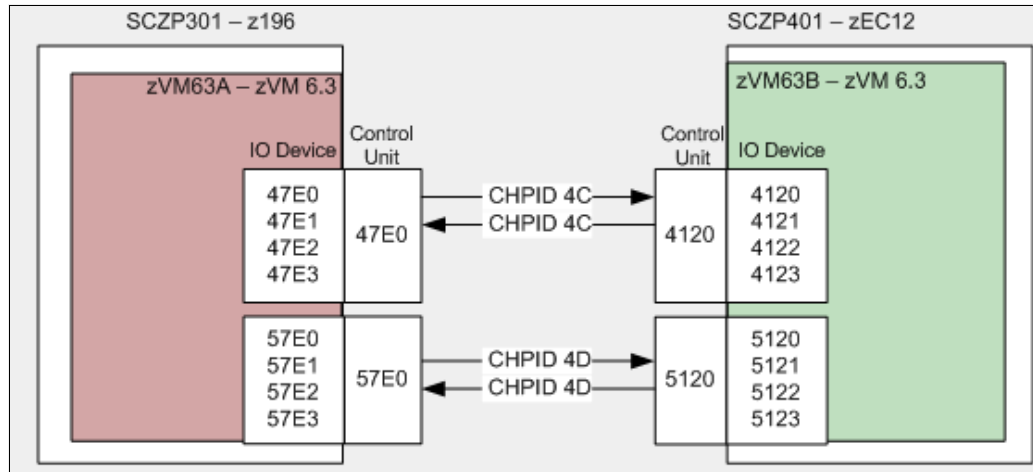


Figure 2-1 FCTCs between the LPARS

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

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 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 *ITS0ZVM1* will be *41A0* and *41A1*. For *ITS0ZVM2* devices, *5190* and *5191* will be used.

For more information about CTC capacity recommendations, see the website:

<http://www.marist.edu/~mvmua/13011100.pdf>

2.11 Architectural overview of the environment that is used in this book

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

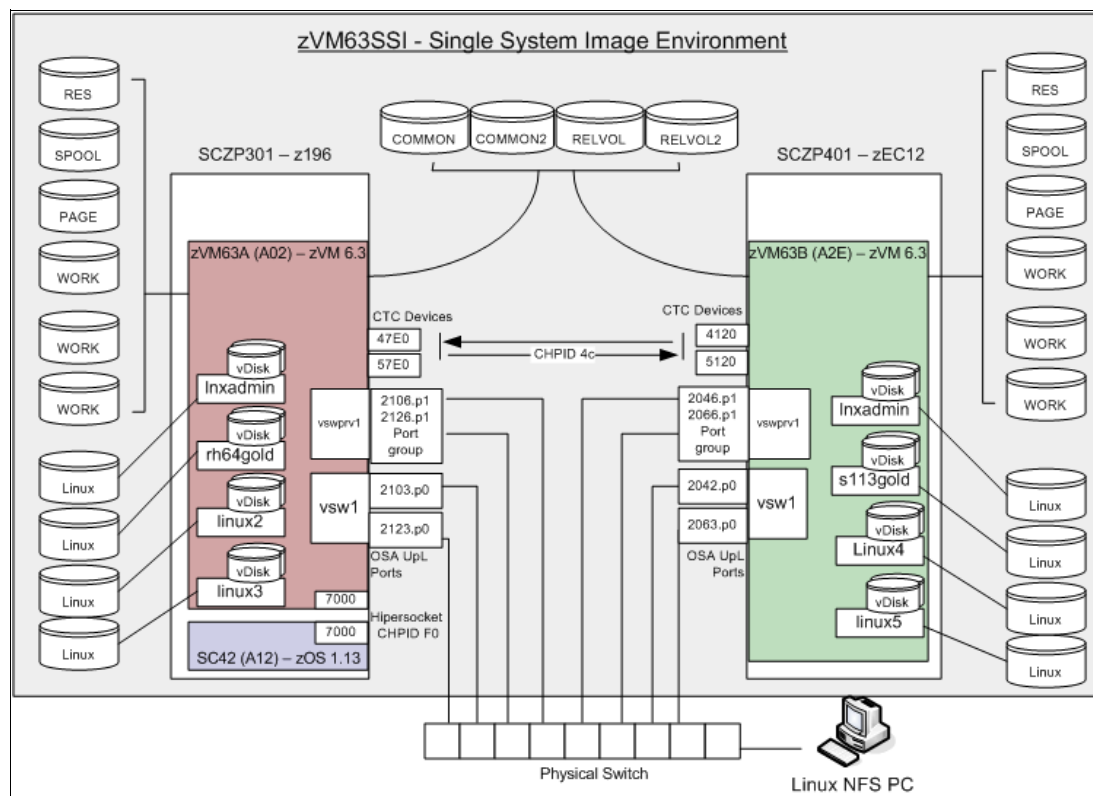


Figure 2-2 Mainframe environment that is used in this book

2.12 Planning worksheet

The following tables make up the overall planning worksheet that is used to install and configure a z/VM 6.3 SSI cluster, Linux guests, and any required supporting resources.

The planning worksheet that is shown here is fully populated with 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 A, “References, cheat sheets, and blank worksheets” on page 307.

Important: The values in the following tables are shown in *monospace bold italics* to signify that you need to replace the example value with the correct value for your site.

2.12.1 IBM Shopz

If you are ordering z/VM by using Shopz as described in 4.1, “Obtaining z/VM through electronic download” on page 52, complete this table to document the values that you will use.

Table 2-2 Shopz data

Name	Value	Comment
Starting URL	http://www.ibm.com/software/info/shopzram/	
User ID	<i>MyCustomerNumber</i>	
Password	<i>MyPassword</i>	
Order number	<i>MyOrderNumber</i>	Write down for reference

2.12.2 Hardware Management Console

In 4.3.1, “Start the z/VM installation” on page 56, you will see how to start a z/VM installation from the Hardware Management Console (HMC). Complete Table A-3 on page 314 to document the values that you will use. Table 2-3 provides information about the values that are required for each row.

Table 2-3 HMC data

Name	Value	Comment
HMC location or URL	https://b01zcihmc104.itso.ibm.com	Physical HMC access or https URL if accessed by the web
HMC user ID	<i>91314801</i>	
HMC password	<i>hmcvm4linux</i>	
FTP source system	<i>9.60.86.10</i>	If you are installing z/VM from an FTP server
z/VM installation directory	<i>/ftp/zvm/vm630</i>	

2.12.3 z/VM Installation Planning Panels (INSTPLAN)

You will need to document the information for your environment before you use INSTPLAN.

INSTPLAN panels 1 and 2

In 4.3.2, “Copy a plain z/VM system to DASD” on page 59, you will see the **INSTPLAN** command, which is run from the Integrated 3270 Console. Table 2-4 provides information about the required values for each row.

Table 2-4 *INSTPLAN values for the first two panels*

Name	Value	Comment
Language	AMENG	AMENG (American English), UCENG (uppercase English), or KANJI (Japan).
DASD model	3390-9	3390 Mod 3 or Mod 9 (FBA disk is not described in this book.).
File pool name	VMPSFS	VMPSFS is the recommended default.
System type	SSI	
Non-SSI system name	-----	Left blank. Non-SSI installation is not described in this book.
Number of members	2	SSI installation only (usually 2 or 4).
SSI cluster name	<i>ITSOSSIA</i>	SSI installation only.
Automatic config?	No	Will z/VM be maintained automatically? “No” is strongly recommended.

INSTPLAN panel 3

Complete this table to document the values that you will use on the third installation panel, which is described in 4.3.2, “Copy a plain z/VM system to DASD” on page 59. The member names will become the z/VM system identifiers, and the LPAR names need to be the same names as on the HMC. Table 2-5 provides information about the required values for each row.

Table 2-5 *INSTPLAN values for panel 3*

Slot	Member name	LPAR name	Comment
1	<i>ITS0ZVM1</i>	<i>A09</i>	Member 1 system identifier and LPAR name
2	<i>ITS0ZVM2</i>	<i>A0A</i>	Member 2 system identifier and LPAR name
3	<i>ITS0ZVM3</i>	<i>A0C</i>	Member 3 system identifier and LPAR name
4	<i>ITS0ZVM4</i>	<i>A0D</i>	Member 4 system identifier and LPAR name

INSTPLAN worksheet 3

Table 2-6 Complete the worksheet in Table A-6 on page 315 to document the volume labels and real device addresses that you will use on the third installation panel that is described at the end of 4.3.2, “Copy a plain z/VM system to DASD” on page 59. Table 2-6 provides

information about the values that are required for each row. *INSTPLAN values for volume*

Type/Purpose	Label	Address	Comment
COMMON	VV ZA0CM1	3031	Common volume
RELVOL	ZAORL1	3034	Release volume
SESVOL	ZA0SS1	3036	VMSES/E service volume
SFSVOL	ZA0SF1	3F34	SFS pool volume
Mem 1 RES	ZA1RS1	3234	Member 1 residence volume
Mem 1 SPOOL	ZA1SP1	3236	Member 1 spool volume
Mem 1 PAGE	ZA1PG1	3334	Member 1 page volume
Mem 1 WORK	ZA1WK1	3336	Member 1 work volume
Mem 2 RES	ZA2RS1	3630	Member 2 residence volume
Mem 2 SPOOL	ZA2SP1	3634	Member 2 spool volume
Mem 2 PAGE	ZA2PG1	3636	Member 2 page volume
Mem 2 WORK	ZA2WK1	372F	Member 2 work volume
Mem 3 RES	ZA3RS1	3831	Member 3 residence volume
Mem 3 SPOOL	ZA3SP1	3834	Member 3 spool volume
Mem 3 PAGE	ZA3PG1	3836	Member 3 page volume
Mem 3 WORK	ZA3WK1	392F	Member 3 work volume
Mem 4 RES	ZA4RS1	3A31	Member 4 residence volume
Mem 4 SPOOL	ZA4SP1	3A34	Member 4 spool volume
Mem 4 PAGE	ZA4PG1	3A36	Member 4 page volume
Mem 4 WORK	ZA4WK1	3B2F	Member 4 work volume

definition

INSTPLAN worksheet 4

The values in Table 2-7 document the common volume and CTC addresses that are used in this book. This pane is shown at the end of 4.3.2, “Copy a plain z/VM system to DASD” on page 59. If only two members exist in the SSI, you need to specify only two pairs of CTCAs (from member 1 to member 2, and vice versa).

Table 2-7 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	5190 5191
To: member 2	41A0 41A1	To: member 2	N/A

2.12.4 z/VM networking resources

Table 2-8 lists the networking resources that are used in the examples in this book. They are needed when you invoke the **IPWIZARD** and when you create a VSWITCH for the Linux virtual machines.

Table 2-8 z/VM and networking resources

Name	Value	Comment
TCP/IP user ID	TCPIP	TCPIP is recommended.
z/VM host name, member 1	<i>itsozvm1</i>	
z/VM host name, member 2	<i>itsozvm2</i>	
TCP/IP domain name	<i>itso.ibm.com</i>	System domain name is usually set in DNS.
TCP/IP gateway	<i>9.12.4.1</i>	The router to and from the local subnet.
DNS server 1	<i>9.12.6.6</i>	Obtain from network administrator.
DNS server 2	<i>9.12.6.7</i>	Obtain from network administrator.
DNS server 3	<i>9.60.70.80</i>	Obtain from network administrator.
Interface name	QDIOETH0	QDIOETH0 is recommended. The fourth character is the letter O . The last character is the numeral 0 (zero).
OSA starting device number	<i>2040</i>	Start of OSA <i>triplet</i> for z/VM TCP/IP stack.
Subnet mask Subnet CIDR mask	<i>255.255.240.0</i> <i>/20</i>	Assigned by your network administrator.
OSA device type	QDIO (layer2)	
VLAN ID		Obtain from network administrator if required.
MTU size	1492	1492 or 8992. Consult the network administrator.
VSWITCH1 primary OSA triplet	<i>4203</i> <i>4204 4205</i>	Specify the first real device number and the next two device numbers will also be used.
VSWITCH1 second OSA triplet	<i>4300</i> <i>4301 4302</i>	Ideally, it needs to be on a different CHPID/OSA.
VMLAN MAC prefix, member 1	<i>02000A</i>	Assigned by your network administrator.
VMLAN MAC prefix, member 2	<i>02000B</i>	Assigned by your network administrator.

Important: For setting the VMLAN MACPREFIX value, *IBM z/VM CP Planning and Administration*, SC24-6178, 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 02xxxx, 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.”

2.12.5 z/VM DASD

Table 2-9 lists the z/VM DASD resource values that are used in the examples in this book.

Table 2-9 z/VM DASD that is used in this book

Device	Label	Type	Notes
1567	VM1567	System (3390-9)	LNXADMIN 100 on member 1 (RHEL)
1568	VM1568	System (3390-9)	LNXADMIN 200 on member 1 (RHEL)
1569	VM1569	System (3390-9)	LNXADMIN 100 on member 2 (SLES)
156A	VM156A	System (3390-9)	LNX: SFS, MONWRITE, LXW minidisks
156B	VM156B	System (3390-9)	LNXADMIN 200 on member 2 (SLES)
156C	VM156C	System (3390-9)	MINIDISKS - POOL1
156D	VM156D	System (3390-9)	MINIDISKS - POOL1
156E	VM156E	System (3390-9)	MINIDISKS - POOL1
156F	VM156F	System (3390-9)	MINIDISKS - POOL1
1222	VV1222	CP owned (3390-3)	LNXADMIN 201 on member 2 (SLES) KIWI
3000		(EDEV 10 GB)	LNXADMIN 150 on member 1 (RHEL)
3001		(EDEV 10 GB)	LNXADMIN 150 on member 2 (SLES)
3002		(EDEV 10 GB)	LINUX1 150 (RHEL)
3003		(EDEV 10 GB)	LINUX2 150 (RHEL)
3004		(EDEV 10 GB)	LINUX3 150 (SLES)
3005		(EDEV 10 GB)	LINUX4 150 (SLES)
3006		(EDEV 10 GB)	
3007		(EDEV 10 GB)	
3008		(EDEV 10 GB)	
3009		(EDEV 10 GB)	

2.12.6 FCP devices

Table 2-10 and Table 2-11 list the z/VM FCP resource values that are used in the examples in this book.

Table 2-10 EDEV LUN assignments

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

Table 2-11 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		
B807 B907	A09	C05076DD9000041C C05076DD90000A7C	4011401700000000	
	A0A	C05076DD9000049C C05076DD90000AFC		

N_Port ID Virtualization (NPIV) device	LPAR	NPIV WWPN	LUN	Used by
B808 B908	A09	C05076DD90000420 C05076DD90000A80	4011401800000000	LNXADMIN on Member 1 (RHEL)
B809 B909	A0A	C05076DD900004A4 C05076DD90000B04	4011401900000000	LNXADMIN on Member 2 (SLES)

2.12.7 Linux resources

Table 2-12 and Table 2-13 list the Linux resources that were used in this book.

Table 2-12 External Linux FTP server resources that were used in this book

Name	Value	Comment
TCP/IP address	9.12.5.251	
User/password	ftuser/linux4vm	
FTP installation source directory	/ftp/linux/rhel/7 /ftp/linux/sles/12	Directory with DVD 1 of each distribution

Table 2-13 Linux common configuration values that were used in this book

Name	Value	Comment
Linux root password	linux4vm	
TCP/IP gateway	9.12.4.1	Obtain from the network administrator.
Subnet mask	255.255.240.0 or /20	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	12345678	Must be eight characters.

2.12.8 Host names and IP addresses

Table 2-14 lists the host names and associated IP addresses that are used in the examples in this book.

Table 2-14 Hosts that are used in this book

Host name	IP address	Virtual machine	Notes
itsozvm1.itso.ibm.com	9.12.5.56	LPAR A09	z/VM 6.3 SSI member 1
itsozvm2.itso.ibm.com	9.12.5.57	LPAR A0A	z/VM 6.3 SSI member 2
vm1nx2-1.itso.ibm.com	9.12.7.96	LNXADMIN	RHEL 7.1 FTP Server
vm1nx2-2.itso.ibm.com	9.12.7.97	LNXADMIN	SLES12 FTP Server
vm1nx2-3.itso.ibm.com	9.12.7.98	LINUX1	RHEL 7.1

Host name	IP address	Virtual machine	Notes
<i>vm1nx2-4.itso.ibm.com</i>	<i>9.12.7.99</i>	LINUX2	RHEL 7.1
<i>vm1nx2-5.itso.ibm.com</i>	<i>9.12.7.100</i>	LINUX3	SLES 12
<i>vm1nx2-6.itso.ibm.com</i>	<i>9.12.7.101</i>	LINUX4	SLES 12
<i>vm1nx2-7.itso.ibm.com</i>	<i>9.12.7.102</i>		
<i>vm1nx2-8.itso.ibm.com</i>	<i>9.12.7.103</i>		
<i>vm1nx2-9.itso.ibm.com</i>	<i>9.12.7.104</i>		
<i>vm1nx2-10.itso.ibm.com</i>	<i>9.12.7.105</i>		
<i>vm1nx2-11.itso.ibm.com</i>	<i>9.12.7.106</i>		
<i>vm1nx2-12.itso.ibm.com</i>	<i>9.12.7.107</i>		
<i>vm1nx2-13.itso.ibm.com</i>	<i>9.12.7.108</i>		
<i>vm1nx2-14.itso.ibm.com</i>	<i>9.12.7.109</i>		
<i>vm1nx2-15.itso.ibm.com</i>	<i>9.12.7.110</i>		



Configuring a workstation for mainframe access

“Technological progress is like an axe in the hands of a pathological criminal.”

— Albert Einstein

This chapter addresses the configuration of a workstation that is running either Linux or Windows to access the mainframe.

The following programs, tools, or utilities will be required for you to successfully access z/VM and Linux from your workstation to complete the tasks that are described in this book.

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.

3.1 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 are using a Windows desktop, you will need a good SSH client. *PuTTY* is perhaps the most commonly used. You can download PuTTY from the following download web page:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

To download from this 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.

Follow these steps:

1. Open PuTTY, and the configuration window that is shown in Figure 3-1 opens. If you spend a few minutes to configure 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** as shown in Figure 3-1 to use the SSH protocol.

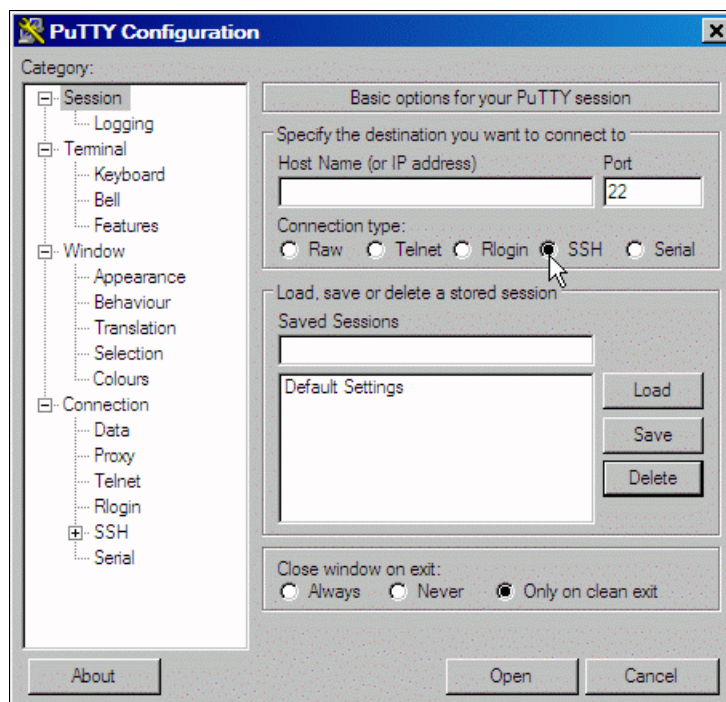


Figure 3-1 PuTTY Configuration window

4. Click **Logging** in the left panel, as shown in Figure 3-2:
 - a. Click **Printable output** in the Session logging radio group to go back and check on the output of certain commands.
 - b. Set the Log file name to `&H&M&D&T.log` so a time stamp will be in the file name.

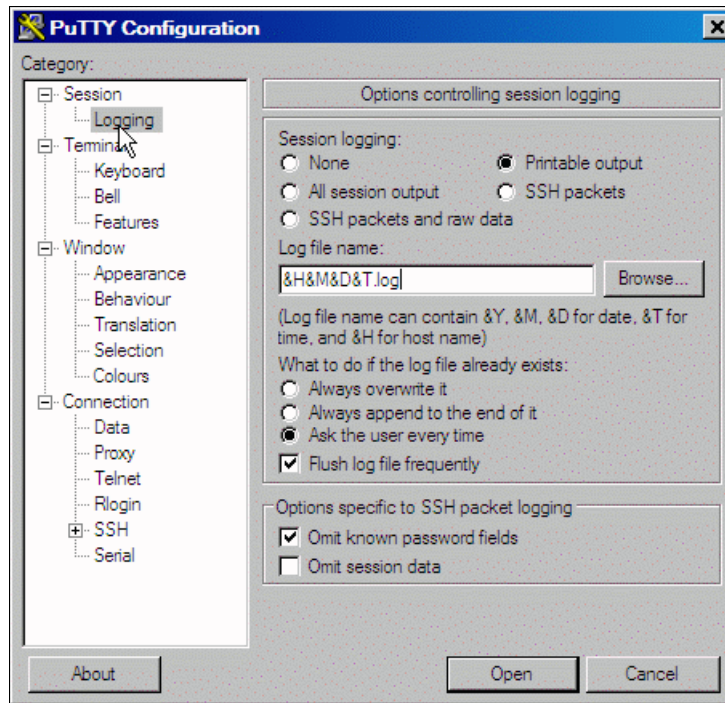


Figure 3-2 Setting logging

5. In the left panel, click **SSH** near the bottom as shown in Figure 3-3.
6. On the right side, under Preferred SSH protocol version, click **2 only**.

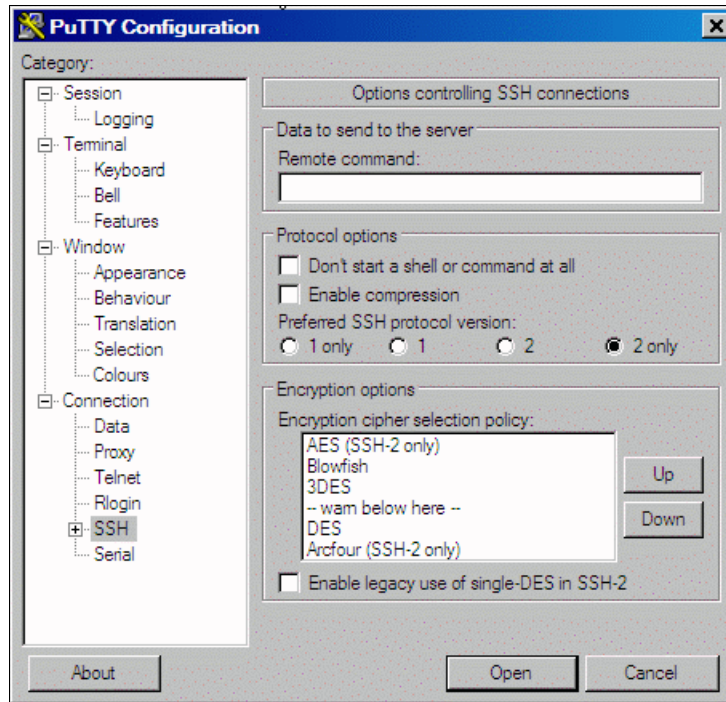


Figure 3-3 Setting SSH Protocol 2

7. In the left Category panel, click **Terminal** as shown in Figure 3-4.
8. Select **Use background colour to erase screen**, which results in a better job of painting the window for applications that use block graphics.

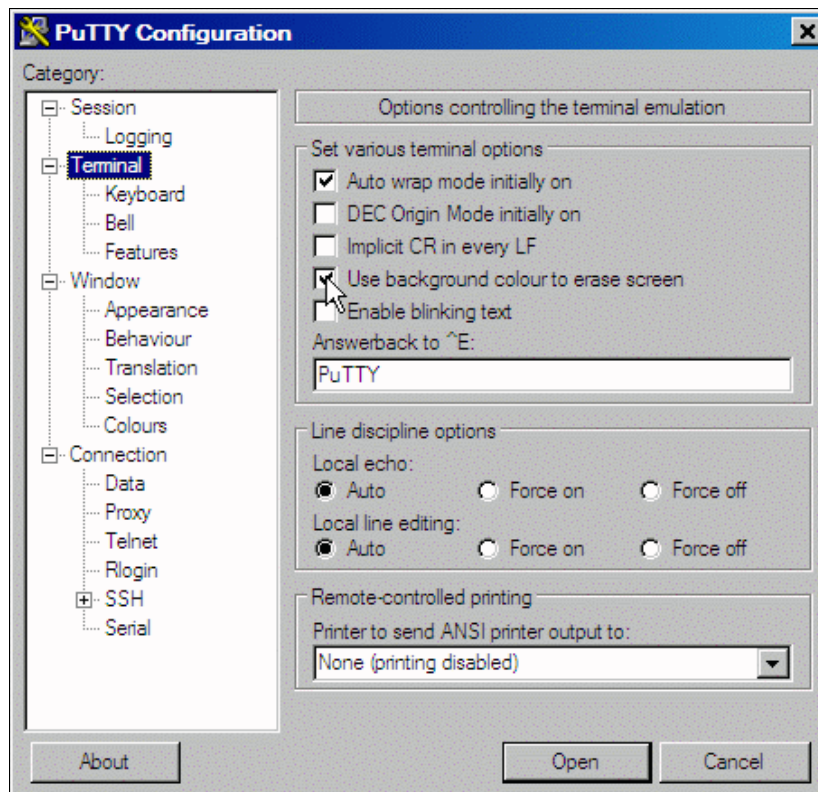


Figure 3-4 Customizing PuTTY SSH settings

9. Click **Window** in the left pane as shown in Figure 3-5.

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.

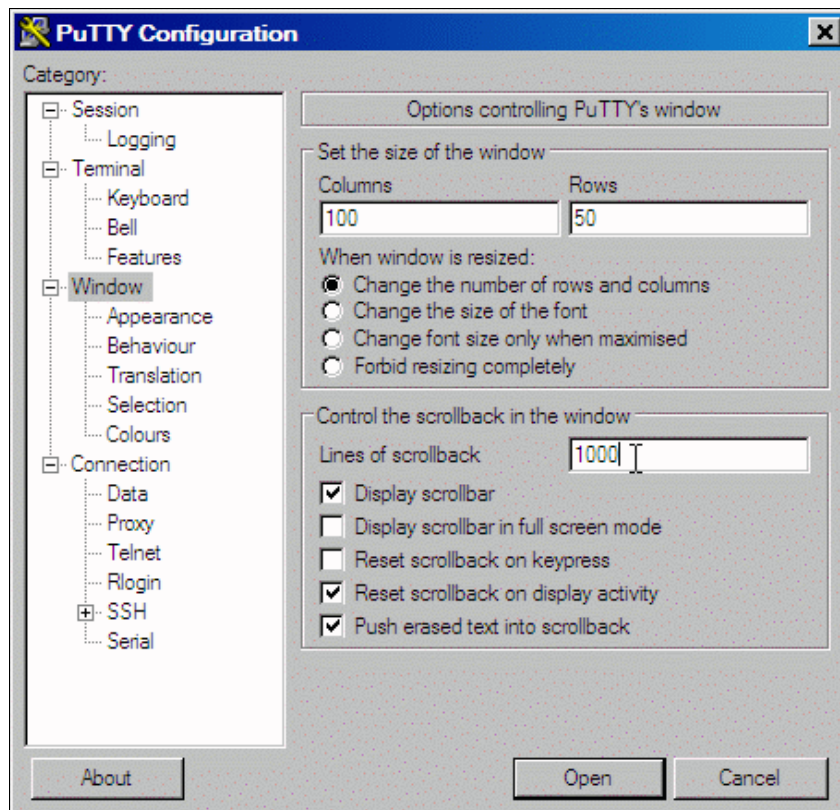


Figure 3-5 Setting window and scrollback size

10. Click **Session** in the left pane as shown in Figure 3-6.
11. Click **Default Settings** in the Saved Sessions pane, then click **Save**. All future sessions that you define will inherit the preferences that you set.

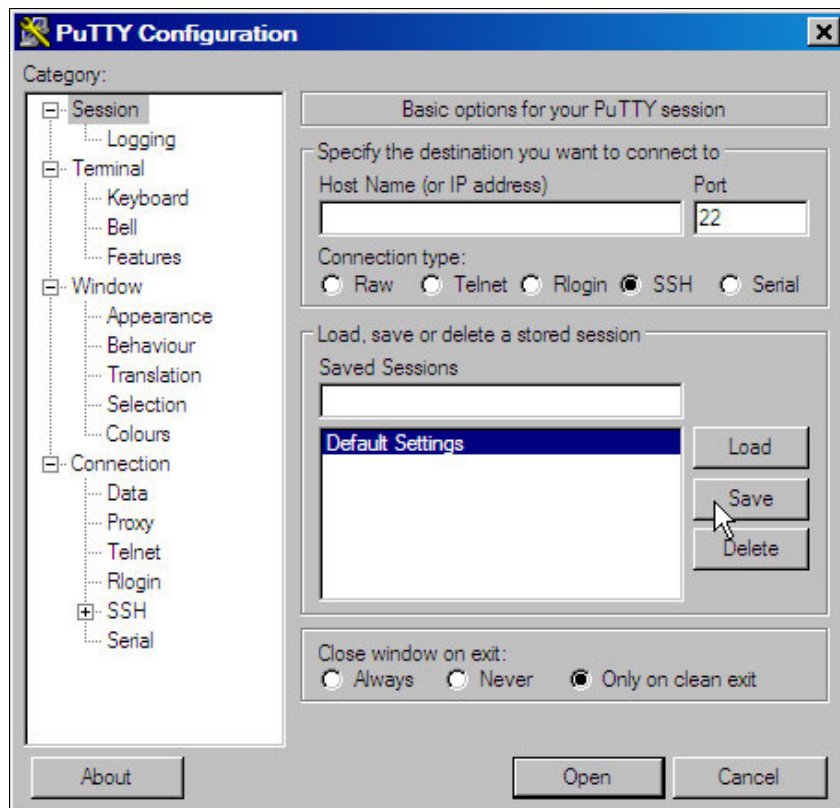


Figure 3-6 Saving the new default settings

Save sessions

To save sessions, perform the following steps. In the example that is shown in Figure 3-7 on page 48, a session for LINUX00 is saved.

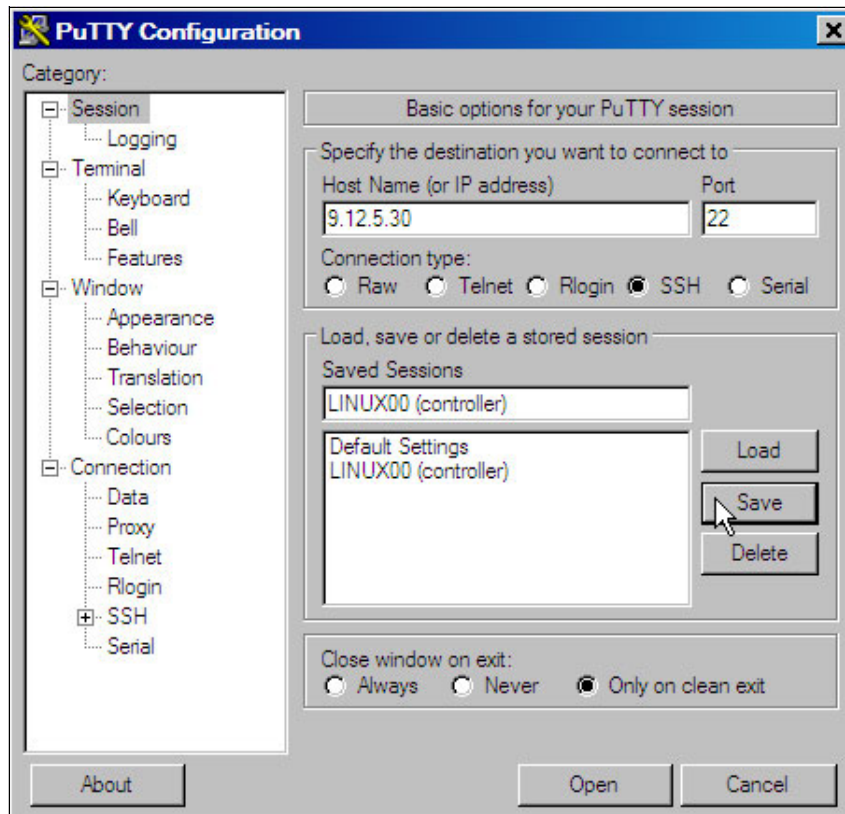


Figure 3-7 Customizing PuTTY window settings

To save a session for each virtual machine, perform 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 will remember. In this example, the name LINUX00 (controller) is used.
3. Again click **Save**, and you see the name added to the Saved Sessions list.

Whenever you start PuTTY, simply double-click any saved session name, and an SSH session to the Linux system that you want will be invoked.

Using a Linux workstation instead is the preferred method.

3.2 Set up a VNC client

A VNC client allows access to a graphical windowing environment with Linux on z.

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. The home page for RealVNC is at the following site:

<http://www.realvnc.com>

The download page is available at the following site:

<http://www.realvnc.com/download.html>

Follow these steps:

1. On the Download 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 probably do not need a VNC server on your desktop. Therefore, you can clear **VNC Server** from the Select Components panel, as shown in Figure 3-8.

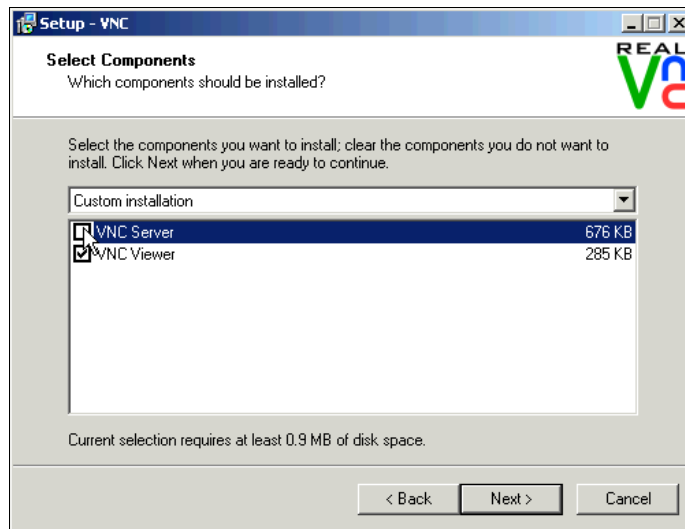


Figure 3-8 RealVNC Select Components panel

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 at the following site:

<http://www.realvnc.com/products/vnc/documentation/5.0/installing-removing/windows>

The tool TightVNC might be an option for the Windows operating systems. See the following site:

<http://www.tightvnc.com/download.php>

The following text is on this page: “TightVNC 2.0.4 supports all client and server versions of Microsoft Windows starting at Windows 2000, up to Windows 7.”

3.3 3270 emulators

To access a logon session with z/VM, it is common to use a 3270 emulator that runs on Windows. Many commercial products are available. The following products are common:

- ▶ Attachmate Extra!
- ▶ Hummingbird Host Explorer
- ▶ IBM Personal Communications
- ▶ Quick3270
- ▶ x3270

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 certain 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. You will probably be more productive with a 43 line window (or more) if they can easily fit in a window based on your desktop display size and resolution.
- ▶ Set up the session to automatically reconnect after logoff. Opening a new logon window automatically immediately after you log off can also save you time. This approach is often not the default behavior.
- ▶ Save your connection sessions. Rather than continually typing the IP address or DNS name of the z/VM system to which you want to connect, spend a few minutes to define and save a session for each system to which you can connect, as was described for PuTTY. Then, you can usually double-click the saved connection to quickly access a new 3270 session.

Customizing your 3270 emulator on the front end can save time later.

3.4 Linux desktops

If you are using a Linux desktop, you need access to these tools:

- ▶ An SSH client, which is named **ssh**. It is part of the **openssh** package.
- ▶ A VNC client, which is named **vncviewer** or **gvncviewer**. It is part of the **tightvnc** package.
- ▶ A 3270 emulator, which is named either **x3270** or **c3270**. It is part of the **x3270** package.



Installing and configuring z/VM

"I hear and I forget. I see and I remember. I do and I understand."

— Confucius

This chapter describes installing z/VM 6.3 as a two-node VM Single System Image feature (VMSSI) cluster, performing 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 VMSSI path. The installation of a single-member SSI cluster provides you with a path for future expansion to add more member nodes later.

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 taken in the previous. Additionally, each chapter and the sections within are specifically sequenced to give you the quickest results.

Notes:

- The steps that are described in detail in this publication are not the official, authoritative installation instructions for z/VM 6.3. The official instruction document entitled, *z/VM Installation Guide, version 6 release 3* is under the *Installation, Migration, and Service* section of the *z/VM V6R3.0 Library* in the IBM Knowledge Center at the following URL:

ibm.com/support/knowledgecenter/SSB27U/welcome

If you are installing exclusively onto Small Computer System Interface (SCSI) disks or by using different options, use the official z/VM documentation.

- Ensure that your z/VM order includes a license for the VMSSI feature. If it does not, contact your IBM marketing representative for details about how to obtain the VMSSI feature.

4.1 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.

To order z/VM, perform the following steps:

1. Visit the *z/VM Service Resources* page:

ibm.com/vm/service

- Under the heading **IBM Support Portals**, click on the link for **IBM Shopz**
2. On the far right of the page, locate the heading **Fast access to Shopz**
 3. If you have not previously used Shopz, choose the appropriate registration link (*IBM Customers* versus *IBM Employees*) to become an enrolled user. If you are already registered, choose the sign-in link that is appropriate for you.
 4. Click **Create new software orders** for service or products.
 5. On *My orders*, click **z/VM Products** and choose **VM SDO version 6** in the drop-down menu to the right. Click **Continue**.
 6. On Step 1 of 7, accept the Order Name and click **Continue**.
 7. On Step 2 of 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. On Step 3 of 7, for the Group, select **VM - VM Base Product**. Select your language, and for the Filter, select **Show all products**. Then, click **Show catalog**. A submenu appears.
 9. Select **z/VM V6 3390 System DDR** and click **Continue**.
 10. On Step 4 of 7, verify the order and click **Continue**.
 11. On Step 5 of 7, verify the entitlements and click **Continue**.
 12. On Step 6 of 7, for the Preferred media, select **Internet** and click **Continue**.
 13. On Step 7 of 7, review and click **Submit**.

It might take time for the order to be prepared. In our example, the email indicating the order was ready for download was received after about two hours. When you receive the email, it contains the URL for downloading your order. Use a browser to go to that URL.

You need to investigate several sections, as shown in Figure 4-1 on page 53. The page has five sections:

- Order Packing List: Contains the list of available products and manuals.
- Installation Instructions: Clicking “View now” takes you to a web page:
ibm.com/vm/install/prodinst.html
- Product Publications: Access different z/VM publications that relate to installation.
- Additional Publications: Download a z/VM SDO document (four pages).
- VM product material: This section is the most important section because it is where you go to download z/VM product installation files. In the example that is used in this book, we clicked **Download to your workstation using IBM Download Director** as

shown in Figure 4-1 on page 53. Download Director is a multi-connection threaded applet and will result in a much faster download than the basic HTTPS method.

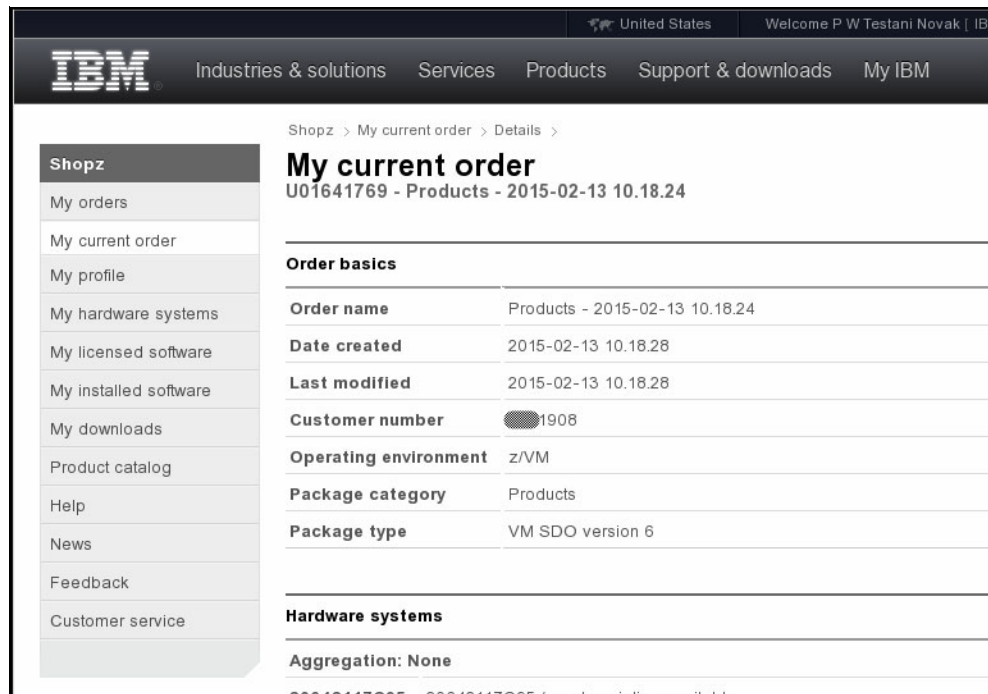


Figure 4-1 Viewing the z/VM order on IBM Shopz

14. Clicking this link displays the window that is shown in Figure 4-2. The second and third check boxes were selected because z/VM 6.3 is being installed onto 3390 DASD. The 1.3 GB of data was downloaded relatively quickly due to multiple connections that are opened by using IBM Download Director.

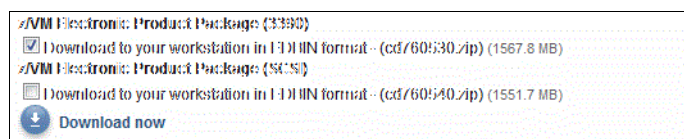


Figure 4-2 Downloading z/VM electronically from IBM Shopz

15. The z/VM installation code will be uploaded to a Linux-based FTP server and unpacked in the next section. In our example environment, the file was staged on a Linux workstation. If you have a pre-existing FTP server you plan to use for the installation of z/VM, you may 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 Installation Tips page:

ibm.com/vm/install/tips

You can now proceed to set up an FTP server.

4.2 Configuring an FTP server for z/VM installation

It is assumed that you can access the z/VM 6.3 installation code in electronic format. Ordering it through IBM Shopz is briefly described in 4.1, “Obtaining z/VM through electronic download” on page 52. If you completed that section, one compressed file that contains the z/VM product installation files might be staged on an intermediate workstation, or you might be ready to download it from the Internet.

Important: Do not unpack or expand the .zip file on a Windows system. Due to the differences in codepages, this can result in corruption of the installation files.

4.2.1 Create 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. Perform 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, either `/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/630/`:

```
# mkdir -p /ftp/zvm/630
```

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. Either 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 IP address `9.60.87.87` 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@9.60.87.87:/ftp/zvm/630
...
```

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/630
# 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 V6R3 is only one compressed file which 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 additional service (updates and fixes) may need to be applied.

7. Next, create directories for the files that are associated with this book. Complete details about these files can be found in Appendix B, “Additional material” on page 321.
8. Create a directory where the files will be stored. In this example, it is /ftp/zvm/cookbook :

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

Note: Previously, this book utilized an example value of /ftp/zvm/sg248147/ for this directory. For user friendliness, the new value shown above has since been adopted. If you had previously created /ftp/zvm/sg248147/, use the following commands to move it and then create a symbolic link for it instead:

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

9. Obtain the latest version of the associated files. You may 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 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: Please visit the following website to verify that you are downloading the latest version of the associated files:

ibm.com/vm/pubs/redbooks/sg248147

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 `maintvm`:

```
# 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, which will be covered in 4.6.3, “Copy the utilities and REXX EXECs to Shared File System (SFS) pools” later in this book.

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

We assume a first-level SSI installation of z/VM from DVD or FTP server onto 3390 DASD. 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 not familiar with the Hardware Management Console (HMC) and z/VM, you might want to use the official z/VM manual, *z/VM 6.3 Installation Guide*, GC24-6246, which is available at this website:

ibm.com/vm/library/zvmpdf.html

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 6.3 Installation Guide*, GC24-6246, instead of this Virtualization Cookbook because we do not address those options.

4.3.1 Start the z/VM installation

The figures in this book show the HMC that uses the default, modern user interface style that is called the *tree view*, not the older *classic view*. The two different user interface (UI) versions are placed side-by-side and shown in Figure 4-3 on page 56.



Figure 4-3 HMC classic view (left) versus tree view (right)

An example of the main menu in tree view mode is shown in Figure 4-4. 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 4-4 HMC tree view

Important: While you work on the HMC, any time that you are prompted for input, such as an IP address, password, parameters, or other value, be aware that **pressing Enter instead of clicking OK is the same as clicking Cancel**.

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

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

1. Log on to the HMC. You will need either physical access to the console or a URL for the web interface. You will need a user ID and password. Assuming that the view is tree mode, you see a window similar to Figure 4-4.
2. Expand **Systems Management** on the left navigation pane. Then, expand **Systems** to view the central processor complexes (CPCs) that are managed by this HMC.
3. Move to the main panel that occupies the right side of your window where the LPARs on which you will install the SSI are shown. Select the LPAR that will be the *first* member of the z/VM 6.3 SSI cluster. The first LPAR (onto which the SSI was installed in this example) is shown in Figure 4-5. 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 absolutely certain that you selected the correct LPAR. If you are not completely sure, check with someone who will know. If you select the wrong LPAR, you irreparably destroy the system that is running on that LPAR.

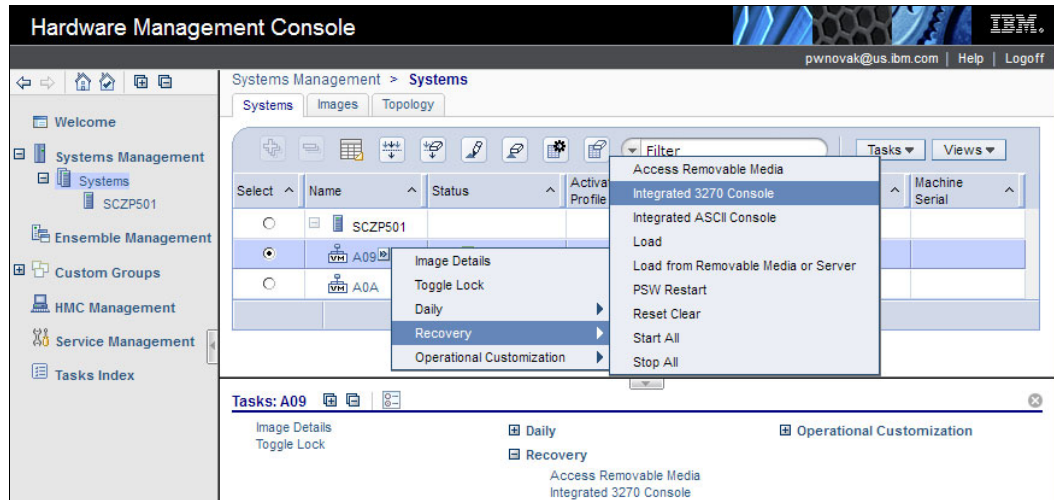


Figure 4-5 HMC with systems selection expanded and Integrated 3270 Console from menu

4. Open an Integrated 3270 Console as shown in Figure 4-5 by clicking the **Tasks** drop-down menu in the upper right, then selecting **Recovery**, followed by selecting **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 4-6 on page 59. Perform the following steps:
 - a. Click **FTP Source**.
 - b. Type the IP address (or host name) of your FTP server into the Host computer field.
 - c. Type the FTP User ID and Password into those fields.
 - d. Type the FTP server directory path that contains 630vm.ims into the File location field. In this example, it is */ftp/zvm/630/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. The FTP Source section is not used.

Load from Removable Media, or Server - ENDHMC1:A09

Use this task to load operating system software or utility programs from a CD / DVD-ROM or a server that can be accessed using FTP.

Select the source of the software:

☐ Hardware Management Console CD / DVD-ROM
☐ Hardware Management Console CD / DVD-ROM and assign for operating system use
☒ **FTP Source**

Host computer:

User ID:

Password:

Account (optional):

File location (optional):

OK Cancel Help

Figure 4-6 Load from Removable Media or Server panel

7. Load the RAMDISK:
 - a. From the Load from Removable Media or Server panel, the directory that contains the file **630VM.INS** is selected. Click **OK**.
 - b. From the Confirm the action window, click **Yes**.
 - c. You see the Disruptive Task Confirmation: Load from CD-ROM, DVD, or Server Progress window. You might be prompted for the password, depending on your HMC configuration.
 - d. 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 after several minutes, you still do not see a message that indicates successful completion, try step 7 again.

An in-memory z/VM 6.3 system is running.

4.3.2 Copy a plain z/VM system to DASD

Follow these steps to copy z/VM to DASD:

1. Move to the Integrated 3270 Console window. The RAMdisk IPLs and you see z/VM boot as shown in Figure 4-7 on page 60. If the Integrated 3270 Console window is still blank, be patient; it can sometimes 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.

```

09:13:49 z/VM V6 R3.0 SERVICE LEVEL 0000 (64-BIT)
09:13:50 SYSTEM NUCLEUS CREATED ON 2013-05-29 AT 10:48:03, LOADED FROM $RAND$
09:13:50
09:13:50 *****
09:13:50 * LICENSED MATERIALS - PROPERTY OF IBM* *
09:13:50 * * *
09:13:50 * 5741-A07 (C) COPYRIGHT IBM CORP. 1983, 2013. ALL RIGHTS *
09:13:50 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE, *
09:13:50 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE *
09:13:50 * CONTRACT WITH IBM CORP. *
09:13:50 * *
09:13:50 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES. *
09:13:50 *****
09:13:50
09:13:50 HCPZC06718I Using parm disk 1 on volume $RAND$ (device FFFF).
09:13:50 HCPZC06718I Parm disk resides on blocks 18000 through 52992.
09:13:50 The directory on volume $RAND$ at address FFFF has been brought online.
09:13:50 HCPWRS2512I Spooling initialization is complete.
09:13:50 No dump unit - Dump function is SET OFF
09:13:50 HCPMLM3016I Management by the Unified Resource Manager is not available
for this system.
09:13:50 HCPAAU2700I System gateway IBMVMRAM identified.
09:13:51 HCPLNM6640E MAINT FFFF not linked. Minidisk has been defined with the
V mode suffix and is already linked by MAINT.
09:13:51 z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
09:13:51 built on IBM Virtualization Technology
09:13:51 There is no logmsg data
09:13:51 FILES: NO RDR, NO PRT, NO PUN
09:13:51 LOGON AT 09:13:51 EDT TUESDAY 04/07/15
09:13:51 SYSG LOGON AS MAINT USERS = 1
09:13:51 HCPI0P952I 2G system storage
09:13:51 FILES: 0000001 RDR, 0000001 PRT, NO PUN
09:13:51 HCPCRC8082I Accounting records are accumulating for userid OPERACCT.
09:13:51 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 V6.3.0 2013-05-30 13:08
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSVLIB does not exist
Ready; T=0.01/0.01 09:13:51

```

RUNNING IBMVMRAM
 42/1

Figure 4-7 First z/VM 6.3 installation window

2. 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)**
3. The command completes quickly and you see the following message:
 HCPDVP8392I: DVDPRIME EXEC ENDED SUCCESSFULLY

- Run the **INSTPLAN DVD** command to set up the configuration for the installation process. You see the z/VM INSTALLATION PLANNING panel as shown in Figure 4-8. In certain cases, you might need to clear the terminal panel by pressing Esc before the INSTPLAN panel appears.

===> **instplan dvd**

- It is recommended that you leave the Ms 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"

  M      VM          M      DIRM          M      ICKDSF
  M      OSA         M      PERFTK         M      RACF
  M      RSCS        M      TCPIP          M      VMHCD

Select a System Default Language.
  X AMENG          _ UCENG          _ KANJI

Select a System DASD model. FBA size can be changed.
  _ 3390 Mod 3      X 3390 Mod 9      _ FBA DASD 6.0

Enter the name of common service filepool.
Filepool Name: VMPSFS

Select a System Type: Non-SSI or SSI (SSI requires the SSI feature)
  _ Non-SSI Install:  System Name _____
  X SSI Install:      Number of Members 2 SSI Cluster Name ITSOSSIA

```

Figure 4-8 z/VM INSTALLATION PLANNING

- Type the letter **X** next to both AMENG (or select your language) and 3390 Mod 9 (or the type of DASD you will use), as shown. The common service file pool default name is **VMPSFS** and it is strongly recommended you use this value.
- Leave the **Non-SSI Install** and **System Name** fields blank. Type the letter **X** next to **SSI Install**, set the number of members (2 in this example), and choose a name for the SSI Cluster Name (**ITSOSSIA** in this example).
- You see the SSI Cluster Installation panel. Fully and completely read the licensing terms to ensure that you understand them, then press **F5** to accept them.
- The z/VM INSTALLATION PLANNING PANEL 2 appears as shown in Figure 4-9 on page 62. 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 typing **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 will create a system which you can not manage with the concepts and process detailed 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 4-9 z/VM INSTALLATION PLANNING PANEL 2

10. You see the z/VM INSTALLATION PLANNING PANEL 3 as shown in Figure 4-10. Enter the SSI 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:  ITS0SSIA

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         ITS0ZVM1         A09
  2         ITS0ZVM2         A0A

```

Figure 4-10 z/VM INSTALLATION PLANNING PANEL 3

11. You will see a summary of your choices and asked if you want to continue. Carefully review all of the values. If the values are correct, type Y to the question and press Enter.
12. You now see the z/VM INSTALLATION VOLUME DEFINITION panel. Initially, it is populated with the default z/VM volume labels VMCOM1, 630RL1, M0... as shown in Figure 4-11 on page 63. You type over these default values with the correct updates from the planning worksheet that you completed. Figure 4-12 on page 63 shows the results of using the example values from the environment that was used to author this book. In this example, a prefix character of V is used. Press F5 to continue.

*** z/VM INSTALLATION VOLUME DEFINITION ***						
TYPE		LABEL	ADDRESS	FORMAT (Y/N)		
COMMON		<u>VMCOM1</u>	_____	—		
RELVOL		<u>630RL1</u>	_____			
TYPE		LABEL	ADDRESS	TYPE	LABEL	ADDRESS
ITS0ZVM1				ITS0ZVM2		
RES		<u>M01RES</u>	_____	RES		<u>M02RES</u> _____
SPOOL		<u>M01S01</u>	_____	SPOOL		<u>M02S01</u> _____
PAGE		<u>M01P01</u>	_____	PAGE		<u>M02P01</u> _____
WORK		<u>M01W01</u>	_____	WORK		<u>M02W01</u> _____

Figure 4-11 z/VM INSTALLATION VOLUME DEFINITION panel with default labels

*** z/VM INSTALLATION VOLUME DEFINITION ***						
TYPE		LABEL	ADDRESS	FORMAT (Y/N)		
COMMON		<u>VV155A</u>	<u>155A</u>	Y		
RELVOL		<u>VV155B</u>	<u>155B</u>			
TYPE		LABEL	ADDRESS	TYPE	LABEL	ADDRESS
ITSOZVM1				ITSOZVM2		
	RES	<u>VV155C</u>	<u>155C</u>		<u>VV1560</u>	<u>1560</u>
	SPOOL	<u>VS155D</u>	<u>155D</u>		<u>VS1561</u>	<u>1561</u>
	PAGE	<u>VP155E</u>	<u>155E</u>		<u>VP1562</u>	<u>1562</u>
	WORK	<u>VM155F</u>	<u>155F</u>		<u>VM1563</u>	<u>1563</u>

Figure 4-12 z/VM INSTALLATION VOLUME DEFINITION panel with planning worksheet values

13. You see the z/VM INSTALLATION FIRST-LEVEL CONFIGURATION panel as shown in Figure 4-13. The common volume addresses will almost always be 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	ITS0ZVM1	ITS0ZVM2	
TYPE	LABEL	ADDRESS	ADDRESS	
=====	=====	=====	=====	
COMMON	VV155A	<u>155A</u>	<u>155A</u>	
CTC device addresses:				
From: ITS0ZVM1			From: ITS0ZVM2	
To: ITS0ZVM1	N/A		To: ITS0ZVM1	<u>5190 5191</u>
To: ITS0ZVM2	<u>41A0 41A1</u>		To: ITS0ZVM2	N/A

Figure 4-13 z/VM INSTALLATION FIRST-LEVEL CONFIGURATION panel

14. Press F5. You see a summary of your values, then the message:

```
...
HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY.
```

15. Reference your planning worksheet and attach all DASD that will be part of the SSI cluster to your virtual machine with 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 1220-1222 155A-1569 156A-156F to *
11:11:23 1220-1222 ATTACHED TO MAINT
11:11:23 155A-1569 ATTACHED TO MAINT
11:11:23 156A-156F ATTACHED TO MAINT
```

Important: The devices **155A-1569 156A-156F 1220-1222** are in bold italics to signify that you need to replace the example values with the correct values from your planning worksheet for your site. This convention is used throughout this book.

16. Run the **INSTALL** command. The DASD will be formatted and the z/VM system disks will be copied. This step usually takes more than one hour:

```
===> install
HCPIIS8490I NOW FORMATTING VOLUME 155A (1 OF ##)
...
```

17. Finally, you see the message HCPMLP8392I **INSTALL EXEC ENDED SUCCESSFULLY**.

Important: It is imperative that the **INSTALL EXEC** succeeds. **IF NOT, YOU MUST NOT PROCEED.** You must go back, fix the issues, and try again.

18. Run the **INSTSCID REMOVE** command to update the SYSTEM CONFIG file:

```
===> instscid remove
...
MSGPFX8392I INSTSCID EXEC ENDED SUCCESSFULLY
```

19. Run the **SHUTDOWN** command. This command shuts down the last SSI 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 **IBMVMMRAM**, which is the in-memory copy of z/VM that was used to begin the installation process.

20. 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 SSI member 1. On the HMC, the LPAR status changes from Operating to Not Operating instead.

z/VM 6.3 is now installed.

4.3.3 IPL the first SSI member

IPL your initial z/VM SSI 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 SSI 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. Follow these 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 4-14. 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 6 RELEASE 3.0					
DEVICE NUMBER	01550	MINIDISK OFFSET:	39	EXTENT:	1
MODULE NAME:	CPLOAD	LOAD ORIGIN:	1000		
-----IPL PARAMETERS-----					
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=155A cons=SYSG					
-----COMMENTS-----					

9= FILELIST 10= LOAD 11= TOGGLE EXTENT/OFFSET					

Figure 4-14 Stand Alone Program Loader

6. At the Start (Warm|Force|COLD|CLEAN) prompt, type **warm** and press the enter key.
==> **warm**
7. At the Change TOD clock prompt, enter no:
==> **no**
8. The first SSI member will IPL cleanly after about a minute. Disconnect from the OPERATOR virtual machine by using the **DISCONNECT** command:
==> **disconnect**

The first SSI member is now running.

4.3.4 IPL for the remaining SSI members

In this example of a two-node SSI cluster, only one more member exists. If you are creating a four-member SSI cluster, you will have three more members.

IPL each of the additional members from the HMC with 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 4-15.

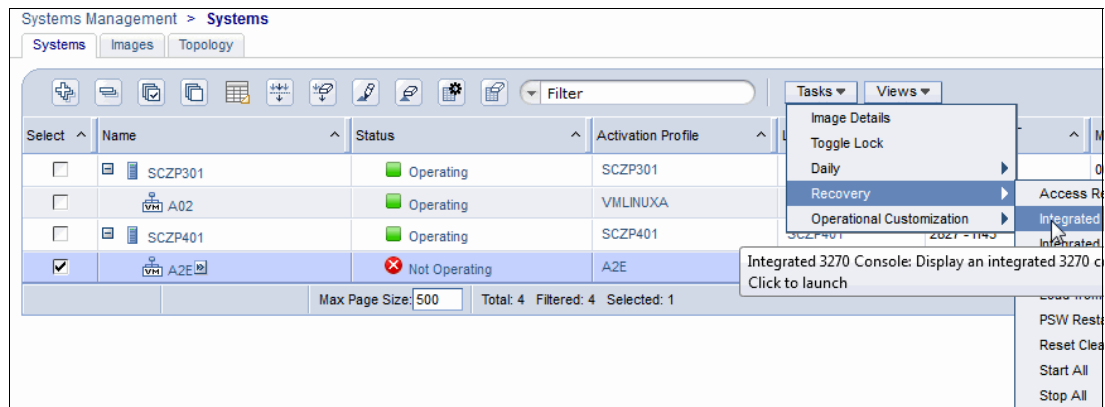


Figure 4-15 Starting a second Integrated 3270 Console

3. Click **Load** on the same Recovery menu. A window opens, as shown in Figure 4-16.
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, type **SYSG**. Click **OK**.

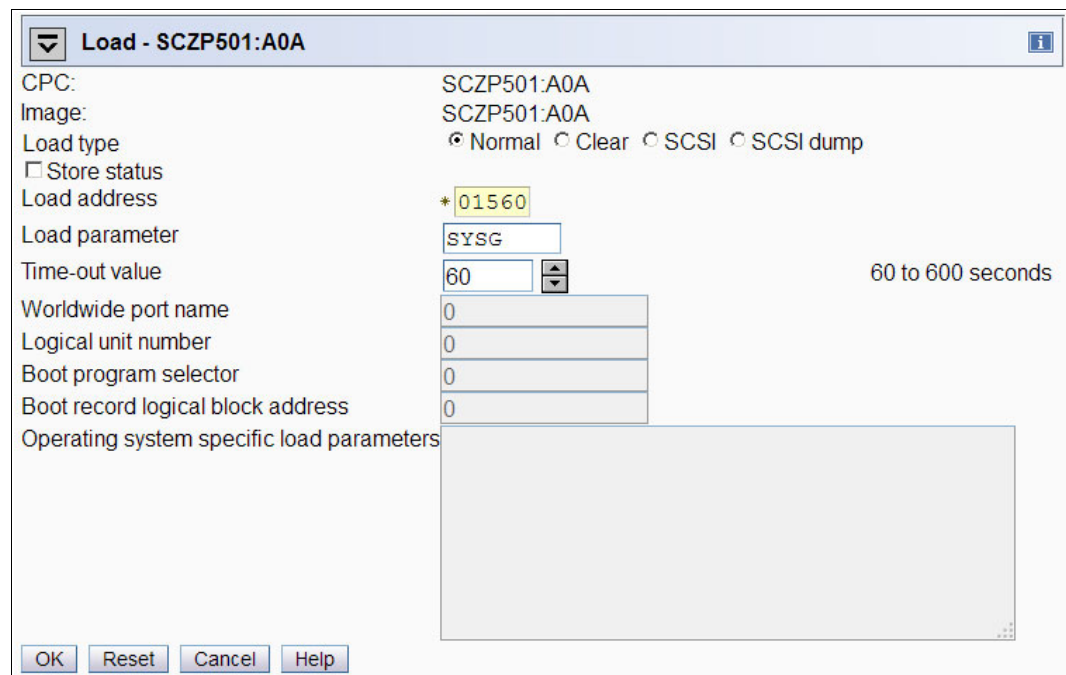


Figure 4-16 Load a second LPAR in the z/VM SSI cluster

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. Watch for important warning below

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 will likely wait indefinitely (forever) 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 typed the CTCs correctly. Figure 2-1 on page 30 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 SSI 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, you will need to troubleshoot your channel-to-channel adapter (CTCA) configuration:

```
==> query ssi
```

```
SSI Name: ITS0SSIA
```

```
SSI Mode: Stable
```

```
Cross-System Timeouts: Enabled
```

```
SSI Persistent Data Record (PDR) device: VV155A on 155A
```

SLOT	SYSTEMID	STATE	PDR HEARTBEAT	RECEIVED HEARTBEAT
1	ITS0ZVM1	Joined	2015-04-09 09:03:01	2015-04-09 09:03:01
2	ITS0ZVM2	Joined	2015-04-09 09:03:08	2015-04-09 09:03:08
3	-----	Available		
4	-----	Available		

10. Invoke 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 SSI cluster are up and running.

4.3.5 Initialize 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 **MAINT630**. The default password for you is provided in the information that you obtained from IBM with your z/VM order.

You will see output similar to the following output:

```
LOGON MAINT630
```

```
z/VM Version 6 Release 3.0, Service Level 1401 (64-bit),  
built on IBM Virtualization Technology
```

```
There is no logmsg data
```

```

FILES: 0004 RDR, NO PRT, NO PUN
LOGON AT 15:12:00 EDT FRIDAY 04/03/15
GRAF L0005 LOGON AS MAINT USERS = 16 FROM 9.60.70.32
z/VM V6.3.0 2015-04-09 09:04

```

```

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 MAINT630 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 MAINT630 and proceed.

4.3.6 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 MAINT630 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 6 Release 3.0, service level 1401 (64-bit)
Generated at 2014-01-21 18:27:52 EDT
IPL at ...

```

3. Visit the RSU page for z/VM on IBM.com to determine the latest available RSU:

ibm.com/vm/service/rsu

- a. Under the column that is labeled RSU Content, click **ZVM630** 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, **1501RSU** tape.

In this example, **1501RSU** 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 5.3, “How to apply a recommended service upgrade” on page 162.
5. Subscribe to the Service News and Red Alerts pages:
 - a. Navigate to the Service News page:

ibm.com/vm/service/news

- 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:

ibm.com/vm/service/redalert

- d. Again, 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**

A z/VM 6.3 SSI cluster exists now. Proceed to begin the configuration of your new SSI cluster.

4.4 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 invoked, it looks for the configuration file `XEDIT PROFILE`. Not all CMS virtual machines always have a copy of this file, so XEDIT sessions can look and behave differently, which can be a problem. The steps in this section will resolve this issue for you.

If you are unfamiliar with XEDIT, a cheat sheet is available in Appendix A, “References, cheat sheets, and blank worksheets” on page 307. Appendix A also includes the URL to the z/VM Library where you can obtain additional 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 both MAINT and MAINT630 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 4-1 shows how to view this basic profile.

Example 4-1 Original MAINT/MAINTvrn 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, perform the following steps:

1. Log on as **MAINT630** on the first SSI member if you are not already.
2. Make a backup copy of 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 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 will 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 you to the command line (====>) at the bottom of the window.
- d. Because our active XEDIT session was launched 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, then verify the result:

```
====> set pf12 ?
```

```
====> query pf12
```

The active definition for PF12 appears at the top of the window as shown in Figure 4-17 on page 71.

```

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 4-17 Output from XEDIT that displays the active definition for PF12

- e. Save the changes up to this point to disk:

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

- f. Enter these 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 command below is **CAPPEND** followed by a space, then a single quotation mark.

```
====> :2
```

```
====> cappend '
```

- h. Use the pound 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 again. 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 may be able to cut and 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 like Example 4-1 on page 70. After you edit it, your **PROFILE XEDIT** looks like Example 4-2.

Example 4-2 Modified MAINT/MAINTvrn XEDIT profile with example date and user ID shown

```
===> type profile xedit
```

```
/*** DEFAULT PROFILE XEDIT FOR z/VM -- MOD 2015-04-06 PWNOKAK ***/
```

```
'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. 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 will not be 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. Don't worry 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
fileid 0017.
====> ip1 190 parm savesys cms
HCPNSS440I Named Saved System (NSS) CMS was successfully saved in fileid
0017.
```
5. LOGOFF as MAINT630 from the current member.
6. **Repeat the previous 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.

4.5 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 need to 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.

Additional subcommands for XEDIT are also described.

4.5.1 Modify features and optimize parameter settings

To make these changes, perform the following steps as MAINT630:

1. Use VMLINK to access the PMAINT CF0 disk as multi-read/write (**MR**) and file mode **F**. You will include the left less than symbol or angle bracket (<) and the right greater than symbol or angle bracket (>) (that are shown) in your command:

```
===> VMLINK PMAINT CF0 < * F MR >
DMSVML2060I PMAINT CF0 linked MR as 0120 file mode F
```

2. Review the existing 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 2015-04-09 09:48:05
```

3. Make a backup copy of the plain SYSTEM CONFIG file by using the **COPYFILE** command with the **OLDDATE** parameter so 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
```

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 2015-04-09 09:48:05
SYSTEM CONFORIG F1 V 80 378 5 2015-04-09 09:48:05
```

5. Open the original file in XEDIT and make the following changes, which are reflected in Example 4-3 on page 76:

```
===> xedit system config f
```

- a. Jump to the line that contains the `set_privclass` statement by using the search (forward slash (/)) subcommand, which works exactly like vi/vim works under Linux:

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


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. In this case, instead of moving to the line that contains `set_privclass`, instead move to the line that contains `auto_warm_ipl`:

```
====> /auto_warm_ipl
```

- b. Add an entry to create an ENABLE stanza:

```
====> input    Enable ,
```

You might choose to append comments to the end of the line that you just added. The resulting line is shown:

```
Enable ,                                /* Enable the following features */
```

- c. Prevent accidental shutdown of the wrong system by enabling validation:

```
====> /Clear_TDisk
====> input      Validate_Shutdown ,
```

- d. Optional: Add an entry to initialize new devices when they are added:

```
====> /Clear_TDisk
====> input      New_Devices_Initialized_When_Added ,
```

- e. Under the RETRIEVE stanza, change `Default 20` to this value:

```
Default 99
```

- f. Under the `PASSWORDS_ON_CMDS` stanza, set all three lines as **no** instead of **yes**:

```
Autolog no ,                          /* ... AUTOLOG does not */
Link    no ,                          /* ... LINK does not */
Logon   no ,                          /* ... and LOGON does not */
```

- g. Set the `Disconnect_Timeout` to **off** so disconnected IDs do not get forced off.

- h. Remove any system-wide limits on VDISK and replace with a 2GB (2G) for each guest user ID limit. The last command in this step does **not** have a trailing comma:

```
====> /Vdisk
====> delete 1
====> input    Vdisk ,
====> input      Syslim infinite ,
====> input      Userlim 2G
```

- i. Add new entries to handle shutdown signals that are used by Linux guests correctly. Jump to the line that contains the string `Userlim` and use the following commands to add these new lines underneath it. Again, the last command in this step does **not** have a trailing comma:

```
====> /Userlim
====> add 1
====> input Set ,
====> input    ShutdownTime 60 ,
====> input    Signal ShutdownTime 500
```

Note: **Signal ShutdownTime 500** will permit any virtual machine that is sent a shutdown signal (sigkill) 500 ticks (wall clock seconds) to complete the shutdown process before it is then forced off. Under most circumstances, this value is more than adequate. **ShutdownTime 60** will permit any virtual machine that is sent a FORCE (forced logoff) 60 seconds to quiesce before the forced logoff happens.

- j. Modify the comments for the lines that you changed, where appropriate. You can use the text that is shown in Example 4-3 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 `*/`, for example:

```
/* DISABLE the following features */
```

Example 4-3 Results of changes to the SYSTEM CONFIG file plus updated comments

```
Features ,
  Disable ,                               /* DISABLE the following features */
    Set_Privclass ,                       /* . . . . SET PRIVCLASS cmd - N */
  Enable ,                               /* ENABLE the following features */
    Auto_Warm_IPL ,                      /* . . . . Auto Warm IPL - Y */
    Clear_TDisk ,                       /* . . Clear TDisk on detach - Y */
    Validate_Shutdown ,                 /* Force Shutdown Validation -- Y */
    New_Devices_Initialized_When_Added , /* AutoInit new devs - Y */
  Retrieve ,                            /* Retrieve options */
    Default 99 ,                        /* . . Orig. IBM default is 20 */
    Maximum 255 ,                      /* . . . Original IBM max is 255 */
  MaxUsers noLimit ,                   /* No limit on number of users */
  Passwords_on_Cmds ,                 /* What commands allow passwords? */
    Autolog no ,                       /* ... AUTOLOG does not */
    Link no ,                          /* ... LINK does not */
    Logon no ,                         /* ... and LOGON does not */
  Disconnect_Timeout off ,             /* Turn off disconnect timeout */
  Vdisk ,                             /* Set VDISK limits ... */
    Syslim infinite ,                  /* ... maximum for system */
    Userlim 2G                        /* ... maximum per user */

Set ,                                  /* Set shutdown and signal values */
  ShutdownTime 60 ,                   /* . . . 60 sec shutdown timeout */
  Signal ShutdownTime 500             /* . . 500 sec wait as sig range */
```

- k. **Save your changes** to this point and proceed:

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

4.5.2 Enable and configure 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 will 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 4.5.1, “Modify features and optimize parameter settings” on page 74. Follow these 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** will turn red. Move down two lines and repeat this step.
 - b. Move to the line after the last statement in the Status of Devices stanza, type **P** into the prefix area and press Enter.
 - c. The entire heading is now duplicated. Example 4-4 shows this duplicated heading.

Example 4-4 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 /*****
```

- d. 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 4-5.

Example 4-5 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, move down two lines and enable INPUT mode:

```
====> /virtual network configuration
====> down 2
====> input
```
4. Type the lines that are shown in Example 4-6 on page 78. Press Enter after each line. Press Enter twice after the last line to return from INPUT mode. Example 4-7 on page 78 shows the entries for both LPARs together with key differences in bold.

Example 4-6 Virtual network configuration values for the first member of the SSI cluster

```
ITS0ZVM1: BEGIN
    RDEV 2100-210F EQID OSA1SET1 TYPE OSA
    RDEV 2120-212F EQID OSA1SET1 TYPE OSA
    VLAN MACPREFIX 02000A
    VLAN LIMIT TRANSIENT 0
    DEFINE VSWITCH VSW1 RDEV 2100 2120 ETHERNET
    MODIFY VSWITCH VSW1 GRANT TCPIP
    DEFINE VSWITCH VSW2 ETHERNET
    MODIFY VSWITCH VSW2 GRANT TCPIP
ITS0ZVM1: END
```

Tip: If you are reading an electronic copy of this book, you can copy the entire block of text from either Example 4-6 or Example 4-7 and paste it directly into your 3270 XEDIT session while you are in INPUT mode. If you are typing manually, carefully enter the text from Example 4-6 and then use the XEDIT block copy operation to duplicate the entire block of text for each additional LPAR in your cluster.

- 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 **VLAN MACPREFIX** statement will set 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. **Ensure that you double-check your work to avoid creating identical MAC addresses.**
- The **VLAN LIMIT TRANSIENT 0** statement prevents dynamic definition of Guest LANs by class G users which will 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 VLAN MACPREFIX value, *IBM z/VM CP Planning and Administration*, SC24-6178, 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 02xxxx, where xxxx is the member's slot number on the SSI statement. If the MACPREFIX value is explicitly defined, the VLAN statement must be qualified for the member to which it applies. Therefore, if a VLAN 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 4-7 Virtual network configuration additions to the SYSTEM CONFIG file

```
/*****
/*
/*          Virtual Network Configuration          */
/*****/
ITS0ZVM1: 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
ITS0ZVM1: END
ITS0ZVM2: 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
ITS0ZVM2: END

```

5. Save your changes and quit XEDIT with the **FILE** subcommand:

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

4.5.3 Use 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 guarantee that it will find all configuration file changes that will 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 postprocessing routines perform further 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 actual target first-level system that the file is intended for, and still might not find all problems that relate to the system configuration file. Follow these 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 MAINT630.

4.6 Initial TCP/IP configuration

It is recommended that you initially configure TCP/IP by using the IPWIZARD command on each of the SSI members. This wizard is generally used only one time. After IPWIZARD creates the initial configuration files, the files are typically maintained manually. A temporary OSA triplet is used to initially get z/VM in the network. Later, the TCP/IP stack will be attached to a highly available z/VM Virtual Switch (VSWITCH) correctly.

4.6.1 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 will need to access it on file mode G by using the **ACCESS** command so that you will pick up **IPWIZARD** from that minidisk. Perform 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 4-18.

- a. The first field, User ID, must always be **TCPIP**.
- b. Obtain the remaining values for your installation from Table A-8 on page 317. Our values are shown in Table 2-8 on page 35. 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: **ITSOZVM1**_____

Domain Name: **ITSO.IBM.COM**_____

Gateway IP Address: **9.12.4.1**_____

DNS Addresses:

1) **9.12.6.6**_____

2) **9.12.6.7**_____

3) **9.60.70.80**_____

Figure 4-18 z/VM TCP/IP Configuration Wizard (panel 1 of 3)

4. Complete the General Interface Configuration Panel that is shown in Figure 4-19 on page 81 with the following information:
 - Set the Interface Name to **QDIOETH0**, which is recommended, as shown.
 - The Device Number is the starting address of the OSA triplet that the z/VM TCP/IP stack will use.
 - The IP Address that must be routed to the OSA card will become the IP address of the z/VM system.

- If you are behind a firewall or other similar type of device that will not permit ICMP Unreachable messages (type 3), do not enable the use of Path MTU Discovery (PMTUD). In all other cases, it needs to 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 will be **QDIO (layer 2)** with modern OSA devices.

When you finish, press **F8**.

```

*** General Interface Configuration Panel ***

Interface Name:  QDIOETH0_____  Device Number:  2103

IP Address:      9.12.5.56_____
Subnet Mask:     255.255.240.0__

Path MTU Discovery (Optional):  X Enabled      _ Disabled

Interface Type (Select one):

_   QDIO (layer 3)      X   QDIO (layer 2)      _   LCS
_   HiperSockets       _   CTC

```

Figure 4-19 General Interface Configuration Panel (configuration wizard panel 2 of 3)

5. In the QDIO Interface Configuration Panel that is shown in Figure 4-20, enter the following information:
 - Enter a virtual LAN (VLAN) ID if your network administrator indicated that a VLAN ID is required.
 - The MTU size for an OSA will be either **1492** or **8992**; this example uses 8992:
 - If your network can support jumbo frames, the use of 8992 provides better performance because OSAs are optimized for this MTU.
 - If you are unable to use the PMTUD feature, the use of 1492 is recommended unless you are positive 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:  8992_

Port Number (optional):  __

```

Figure 4-20 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 given are correctly associated.

4.6.2 Configure TCPIP to automatically start during the system IPL

Follow these steps:

1. Use VMLINK to access the 191 disk for AUTOLOG1 in FILELIST:

```
====> vmlink autolog1 0191 < * * MR > (filelist
```

2. FILELIST will launch 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. Jump to the line that begins with Customer processing, move down one line, and insert an XAUTOLOG statement for the TCPIP ID:

```
====> /Customer processing
```

```
====> down 1
```

```
====> input "PIPE CP XAUTOLOG TCPIP"
```

4. Your results need to look like this example:

```
/* **** */
/* Customer processing can be added here */
/* **** */
"PIPE CP XAUTOLOG TCPIP"
```

5. Save your changes and quit XEDIT:

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

6. You are now returned back to FILELIST. Press PF3 to quit FILELIST and return to CMS. VMLINK will then automatically release and detach AUTOLOG1 191 for you:

DMSVML2061I AUTOLOG1 0191 detached

7. **LOG OFF** so that the PMAINT 2CC disk is freed.

Important: For all other members in the SSI cluster, you must now repeat all of 4.6, "Initial TCP/IP configuration" on page 80. When you run IPWIZARD on the other members, you will 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 SSI 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 3.3, “3270 emulators” on page 49.

To switch to a 3270 emulator, ensure that you issued **LOGOFF** from any Integrated 3270 Console sessions that might still be open.

4.6.3 Copy the utilities and REXX EXECs to Shared File System (SFS) pools

Perform the following steps:

1. Log on as **MAINT630** on the first member from your terminal emulator.
2. Issue the following command to create a new directory underneath the MAINT630 ID in the system-generated clustered product SFS pool, VMPSFS:

```
====> CREATE DIRECTORY VMPSFS:MAINT630.UTILS
```

Note: We are using the VMPSFS filepool as a staging area because VMPSFS is created at the VMSSI cluster level. We will move items out of this filepool into either the VMSYSU filepool on each member, or into the LNX filepool that we will create later in this chapter. You should 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:MAINT630.UTILS. M (forcerw)
```

6. On the FTP server, the directory path /ftp/zvm/cookbook/maintvrn (or /ftp/zvm/sg248147/maintvrn) was created automatically through the expansion of the .tgz file in 4.2.1, “Create directories on the FTP server and upload the installation image” on page 54. We are now ready to transfer files from the FTP server to the VMPSFS file pool using one of the following methods; *either a or b*:

- a. 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/maintvrn
====> 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...

```

a. Or, perform a **put** (push) from the FTP server using the following steps:

- i. Start by first issuing the following command while logged on as **MAINT630** to any member of the z/VM cluster:

```
===> enroll administrator ftpserve vmppsfs
```

This will temporarily grant administrative access to this SFS pool for the virtual machine running the z/VM FTP server. Without this permission, you would not be able to issue the **cd** command from an FTP session.

- ii. Next, log on to your FTP server and initiate an FTP session to z/VM:

```

$ ftp itsozvm1.itso.ibm.com
Connected to itsozvm1.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 (itsozvm1.itso.ibm.com:pwnovak):
maint630
331 Send password please.
Password:
230 MAINT630 logged in; working directory = MAINT630 191
Remote system type is z/VM.
cd VMPSFS:MAINT630.UTILS
ftp> cd VMPSFS:MAINT630.UTILS

```

Note: At this point, if you had skipped the administration enrollment from the previous step, you would see the following:

```

550 FTP server does not have administrator authority for this file
pool; directory remains MAINT630.191

```

```
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): maint630
331 Send password please.

```

```

Password:
230 MAINT630 logged in; working directory = MAINT630 191
Remote system type is z/VM.
ftp> cd VMPSFS:MAINT630.UTILS

```

```

===>
===> mput *
===> quit

```

7. Check the listing of the files that you just downloaded into SFS:

```

===> vmfclean # 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:MAINT630.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 MAINT630 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:MAINT630.UTILS. < . M * > (NON'  
'SET FILEPOOL ...
```

Each SSI member now has access to the **CALLSM1**, **CPFORMAT**, and **SSICMD** EXECs.

4.7 Adding additional 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 planning” on page 26.

4.7.1 Format 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 A, “References, cheat sheets, and blank worksheets” on page 307. 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 2.3.1, “DASD volume labeling convention” on page 20. 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 12, “Miscellaneous helpful information” on page 299.

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.

4.7.2 Use the CPFORMAT EXEC

To use the **CPFORMAT** EXEC, perform 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 4.6.3, “Copy the utilities and REXX EXECs to Shared File System (SFS) pools” on page 83):

```
===> filelist
```

```
MAINT FILELIST A0 V 169 Trunc=169 Size=3 Line=1 Col=1 Alt=0
```

```
Directory = VMPSFS:MAINT630.UTILS
```

Cmd	Filename	Filetype	Fm	Format	Lrec1	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 parameter of a question mark (?):

```
===> 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 A-9 on page 317. Our values are shown in Table 2-9 on page 36.

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 will be 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, ITS0ZVM2.

4.7.3 Format 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. Invoke 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.

4.7.4 Update 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, perform the following steps as MAINT630:

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 2015-04-09 09:56:31
SYSTEM CONFORIG F1 V 80 378 5 2015-04-09 09:56:31
-1SYSTEM CONFORIG F1 V 80 378 5 2015-04-09 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 4-21 and after the modification in Figure 4-22 on page 90.

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

    ITS0ZVM1:  BEGIN
                CP_Owned   Slot 255  VP155E
    ITS0ZVM1:  END

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

    ITS0ZVM2:  BEGIN
                CP_Owned   Slot 255  VP1562
    ITS0ZVM2:  END
```

Figure 4-21 SYSTEM CONFIG file before modification

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

    ITS0ZVM1:  BEGIN
                CP_Owned   Slot 253  VP155E
                CP_Owned   Slot 254  VP1565
    ITS0ZVM1:  END

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

    ITS0ZVM2:  BEGIN
                CP_Owned   Slot 253  VP1562
                CP_Owned   Slot 254  VP1566
    ITS0ZVM2:  END
```

Figure 4-22 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 4-8, all volume labels that begin with **VM15** will be attached to SYSTEM and be available for the creation of minidisks.

Example 4-8 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. In this example, the list looks like this list:

```

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.

4.7.5 Attach minidisk volumes to the system for use

Detach the volumes for minidisks from MAINT and attach them to SYSTEM as shown:

```

====> 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

```

...

4.8 Enabling the IBM Directory Maintenance Facility (DirMaint)

Note: IBM 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.

To add a user to z/VM, you must create a directory entry for a 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. The USER DIRECT file is a CMS file that contains the configuration values and the values of the settings for each guest virtual machine.

A virtual machine definition is a grouping of directory statements that begin with the term USER or IDENTITY. For additional details, a description of virtual machine types, and the USER and IDENTITY statements, see *z/VM Getting Started with Linux on System z*, SC24-6194.

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:

- ▶ For the VMSSI feature to function correctly, and relocation to function as designed, the user directories on all members of an SSI cluster must always remain synchronized.
- ▶ 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 VMSSI cluster, it is illogical to attempt manual user management when automation exists.

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.

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

Next, the initialization of DirMaint for directory management across the cluster is explained.

4.8.1 Enable DirMaint

You need to 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 ships preinstalled with z/VM in a disabled state. To enable it, perform the following steps:

1. Log on as **MAINT630** on member 1 of the VMSSI cluster.

Important: You must use the **MAINT630** 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 MAINT630's link.

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
VMFP2P2760I PUT2PROD processing started for DIRM
VMFP2P1233I The following products have been put into production. Recycle the appropriate servers.
VMFP2P1233I DIRM
VMFP2P2760I PUT2PROD processing completed successfully.

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.

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
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
===> type system config z
...      // several panels of text cleared...

PRODUCT PRODID 6VMDIR30 STATE ENABLED DESCRIPTION '04/09/15.12:18:34.MAINT630
Install/service DirMaint using minidisk'
===> release z (detach
```

Tip: You can also perform this step in only one command instead of three commands. Note that the parenthesis are part of the command and must be included:

```
===> vmlink pmaint cf0 (invoke type system config z
```

6. Log off from MAINT630 on node 1.
7. Log on to the next node in the cluster as **MAINT630** and put DirMaint into production on that node by issuing the **PUT2PROD DIRM** command. Repeat this action for each node in the cluster.

DirMaint is now enabled across all members of the SSI cluster.

If you want, you can check the DIRMAINT and DIRMSAT# virtual service machines (VSMs) across all member nodes by using these commands:

```
===> query DIRMAINT at all
===> query DIRMSAT1 at all
```

4.8.2 Change default passwords

To take the first major step toward correctly securing your new z/VM system, perform the following steps:

1. Change the default passwords for Service virtual machines (SVMs):
 - a. Log on as **MAINT630** on the first member of the SSI cluster.
 - b. 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 at all, 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
```

- c. Open the z/VM user directory for editing:

```
====> xedit user direct C
```

- d. Change the passwords of 6VMDIR30, DIRMAINT, DIRMSAT, DIRMSAT_x (where _x is 2, 3, or 4, depending on the number of SSI member nodes), DATAMOVE, and DATAMOV_x from their current value (typically AUT0ONLY) to an eight character value of your choice. These IDs are powerful, so choose non-trivial values.

```
====> /user 6vmdir30
USER 6VMDIR30 NEWPASWD 16M 64M EG
```

```
====> top
====> /user dirmaint
...
```

- e. 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 **MDRKI90P**. This command is case-sensitive, so enter it by using all uppercase letters:

```
====> C/MDRKI90P/NEWPASWD/**
DMSXCG517I ## occurrence(s) changed on ## line(s)
```

- f. Run the **DIRECTXA** command as MAINT630 on all members to bring the changes online:

```
====> DIRECTXA USER DIRECT
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0

EOJ DIRECTORY UPDATED AND ON LINE

HCPDIR494I User directory occupies 107 disk pages
```

2. Log on as **6VMDIR30** on the first member of the SSI cluster by using the new password that you set.

3. Access the 492 disk as E to get access to the **DIR2PROD EXEC**:

```
====> access 492 E
```

4. Use the **DIR2PROD EXEC** to access the necessary minidisks:

```
====> dir2prod access_new 6vmdir30 dirm
DMSACP726I 492 E released
DIR2PROD: Normal Termination.
```

5. Three new minidisks were accessed as J, K, and L:

```
====> query accessed
```

Mode	Stat	Files	Vdev	Label/Directory
A	R/W	2	191	DRM191
B	R/O	133	5E5	MNT5E5
D	R/W	299	51D	MNT51D
J	R/W	65	1DF	DIR1DF
K	R/W	284	492	DRM492
L	R/W	55	41F	DRM41F
S	R/O	698	190	MNT190
Y/S	R/O	1124	19E	MNT19E

6. To access the user directory source statements, link to the MAINT 2CC disk read-only with the **VMLINK** command. The read password will either be the value that you set all passwords to, or if you did not change them, it will be **READ**:

```
====> vmlink maint 2CC
ENTER READ PASSWORD:
```

DMSVML2060I MAINT 2CC linked as 0120 file mode Z

7. Copy the USER DIRECT file from MAINT 2CC (file mode Z) to DIRMAINT 1DF (file mode J) as the file USER INPUT, which will cause the current user directory to be loaded into DirMaint when it starts for the first time:

```
===> copy user direct Z = input J
```

8. Create the primary DirMaint local customization parms 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 4-9. 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
```

```
...
```

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

- The ALLOW_ASUSER_NOPASS_FROM lines enable z/VM Systems Management API (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.
- 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 will not create a VMBATCH-specific resource entry. Otherwise, DIRMAINT will create a VMBATCH resource for this user ID with this line as a default. The VMBATCH generic resource class is configured in 8.2, “Enable and configure RACF” on page 217. If you are not installing RACF, you can omit this line.

Example 4-9 Example CONFIGAA DATADVH file contents

```
ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKS *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKC *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKL *
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
DISK_CLEANUP= YES
PW_INTERVAL_FOR_SET= 90
ONLINE= IMMED
RUNMODE= OPERATIONAL
RACF_RDEFINE_VMBATCH_DEFAULTS=
```

9. Create the AUTHFOR CONTROL file on the J disk (DIRMAINT 1DF). Add 22 lines to accommodate the entries that are shown in Figure 4-23 for all IDs that are required to perform DirMaint tasks.

```
===> xedit authfor control j  
====> a 22
```

```
ALL LNXADMIN * 140A ADGHOPS  
ALL LNXADMIN * 150A ADGHOPS  
ALL LNXMAINT * 140A ADGHOPS  
ALL LNXMAINT * 150A ADGHOPS  
ALL MAINT630 * 140A ADGHOPS  
ALL MAINT630 * 150A ADGHOPS  
ALL MAINT * 140A ADGHOPS  
ALL MAINT * 150A ADGHOPS  
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  
ALL WAVEWRKS * 140A ADGHMOPSZ  
ALL WAVEWRKS * 150A ADGHMOPSZ  
ALL WAVEWRKL * 140A ADGHMOPSZ  
ALL WAVEWRKL * 150A ADGHMOPSZ  
ALL WAVEWRKC * 140A ADGHMOPSZ  
ALL WAVEWRKC * 150A ADGHMOPSZ
```

Figure 4-23 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 positive that you will not use IBM Wave to graphically manage your system, you can omit the WAVEWRK* lines.

Many of the DirMaint configuration files are now created. The next important file is the EXTENT CONTROL file.

4.8.3 Customize 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 need to 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, perform 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 155A CP OWNED VV155A 36
DASD 155B CP SYSTEM VV155B 66
DASD 155C CP OWNED VV155C 211
DASD 155D CP OWNED VS155D 1
DASD 155E CP OWNED VP155E 0
DASD 155F CP SYSTEM VV155F 2
DASD 1561 CP OWNED VS1561 0
DASD 1567 CP SYSTEM VV1567 1
DASD 1568 CP SYSTEM VV1568 1
DASD 156A CP SYSTEM VM156A 9
DASD 156C CP SYSTEM VM156C 0
DASD 156D CP SYSTEM VM156D 0
DASD 156E CP SYSTEM VM156E 0
DASD 156F CP SYSTEM VM156F 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** to 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. If you are not sure of the device type, use the **QUERY DASD DETAILS <rdev>** command from **MAINT**. Each region name is also added to one or more **GROUPS**.

DirMaint provides the capability to clone a **SUBCONFIG** entry by using an existing **SUBCONFIG** entry. With 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.

Important: As you enter your **Dev-Type** values, you **must** use two digits for the **Type**. 3390 model 3 regions must be entered as 3390-03, 3390 model 9 regions must be entered as 3390-09.


```

==> xedit extent control j
* ****
...
Purpose:  Default Extent Control file.
...
* ****
:REGIONS.
*RegionId  VolSer    RegStart    RegEnd  Dev-Type  Comments
VV155B    VV155B    0001        END     3390-09
VV155F    VV155F    0001        END     3390-09
VV1567    VV1567    0001        END     3390-09
VV1568    VV1568    0001        END     3390-09
VV1569    VV1569    0001        END     3390-09
VM156A    VM156A    0001        END     3390-09
VV156B    VV156B    0001        END     3390-09
VM156C    VM156C    0001        END     3390-09
VM156D    VM156D    0001        END     3390-09
VM156E    VM156E    0001        END     3390-09
VM156F    VM156F    0001        END     3390-09
VM1222    VM1222    0001        END     3390-03
:END.
:GROUPS.
*GroupName RegionList
* SYSTEM is for z/VM System Volumes
SYSTEM    VV155B
* USRWORK is for z/VM Work Volumes on all members
USRWORK   VM155F VM1563
* LNXADM1 is for full-pack minidisks used by LNXADM-1
LNXADM1   VM1567 VM1568
* LNXADM2 is for full-pack minidisks used by LNXADM-2
LNXADM2   VM1569
* POOL1 is for Linux virtual machines
POOL1     VM156A VM156B VM156C VM156D
POOL1     VM156E VM156F
* POOL2 is for Kiwi
POOL2     VM1222
:END.
:SSI_VOLUMES.
* Added during Installation, Do not remove.
*VolumeFamily  Member  VolSer
IBM_RES        ITS0ZVM1 VV155C
IBM_WORK1      ITS0ZVM1 VM155F
IBM_RES        ITS0ZVM2 VV1560
IBM_WORK1      ITS0ZVM2 VM1563
:END.
:DEFAULT_GROUPS.
*GroupName Member
: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.

```

4. Update the DirMaint configuration:

```

===> dir2prod update_files 6vmmdir30 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.

```

5. Copy CONFIGAA DATADVH to the 11F minidisk:

```

===> ACCESS 11F F
===> ACC 41F L
===> COPY CONFIGAA DATADVH L = = F (OLDDATE

```

6. Log off from 6VMDIR30.

The EXTENT CONTROL file, which is read when DirMaint starts, is now configured.

4.8.4 Start DirMaint

To start DirMaint, perform the following steps:

1. Log on as MAINT on the first SSI member.
2. Issue the following command, which is really 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 ENDM363. - 2015/04/14; T=0.01/0.01 22:02:15
DIRMAINT: DVHILZ3510I Starting DVHINITL with directory: USER INPUT E
DIRMAINT: DVHILZ3510I DVHINITL Params: BLDMONO NOCRWARN
...
DIRMAINT: DVHWAI2140I Waiting for work on 15/04/14 at 22:03:45.

```

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, CONFIGAA DATADVH, AUTHFOR CONTROL, and EXTENT CONTROL configuration files.

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.

4.8.5 Validation of DirMaint

To validate your DirMaint installation, perform the following steps:

1. Disable the prompt for re-entry of the password for each DirMaint command. Send the **needpass no** command and type the MAINT password. Watch for a 0 return code:

```

===> dirmaint needpass no
DVHXMT1181R Enter the current logon password of MAINT at ITS0ZVM1 for
DVHXMT1181R authentication. It will not be displayed on the terminal.
DVHXMT1181R To exit without processing the command, just press ENTER.
... // You will be asked to type the password for MAINT twice
DVHXMT1191I Your NEEDPASS request has been sent for processing to DIRMAINT

Ready;
DVHREQ2288I Your USEROPTN request for MAINT at * has been accepted.
DVHBIU3450I The source for directory entry MAINT has been updated.
DVHBIU3456I Object directory update is not required for this source update.
DVHREQ2289I Your USEROPTN request for MAINT at * has completed; with RC = 0.

```

If you do not get a 0 return code, go back to review your configuration settings.

Note: This step disables only the prompt from MAINT. It is not a system-wide change.

2. Optional: Issue these commands to update the terminal characteristics for MAINT and MAINT630 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 maint630 COMMAND ADD 10 SCREEN STAT RED REV**

====> **dirmaint for maint630 COMMAND ADD 10 SCREEN INAR YEL UND**

3. Issue 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.

4. 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 type `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.

5. 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

6. Query DirMaint for the listing of DASD groups that were defined in EXTENT CONTROL:

====> **dirmaint DASD QUERY GROUP ***

7. Query the current STORAGE values that were set for DIRMAINT:

====> **dirmaint for dirmaint storage ?**

...

DVHST03207I DIRMAINT currently has a maxstorage value of **256M** and a

DVHST03207I default storage value of **128M**.

DVHREQ2289I Your STORAGE request for DIRMAINT at * has completed; with RC = 0.

8. 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 ITS0ZVM1 Sysaffin: * Activity: INACTIVE Pending: 0
DVHSTT3418I CurUnit: Autolog Attempts: 0
DVHSTT3418I DATAMOV2 ITS0ZVM2 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.

4.9 Implementing additional 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.

4.9.1 Enable z/VM FTP and Network File System functionality

Enable both the FTP and Network File System (NFS) functions by performing the following steps:

1. Log on to **TCPMAINT** by using the password that you set for service machines in 4.8.2, “Change default passwords” on page 94.
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. As shown in Example 4-11 on page 104, type **LGLOPR WAVEWRKS WAVEWRKL**.
 - d. Locate the last line that begins with **; 2049** and move your cursor into the prefix area beside the line underneath that begins with **; -----**
 - e. Enter **I3** into the prefix area and press Enter to add three blank lines.
 - f. As shown in Example 4-11 on page 104, type an **AUTOLOG** statement, a middle line for **FTPSEVER X** that logs on the FTP server when TCP/IP starts, and the **ENDAUTOLOG** statement.
 - g. In the **PORT** stanza, remove the semicolons to uncomment the lines with **FTPSEVER** 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 4-10 on page 104.

Example 4-10 Initial PROFILE TCPIP file

```
...
; -----
OBEY
OPERATOR TCPMAINT MAINT MPROUTE REXECD SNMPD SNMPQE LDAPSRV MAINT630
ENDOBAY
; -----
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 4-11.

Example 4-11 Modified PROFILE TCPIP file

```
...
; -----
OBEY
OPERATOR TCPMAINT MAINT MPROUTE REXECD SNMPD SNMPQE LDAPSRV MAINT630
LGLOPR WAVEWRKS WAVEWRKL
ENDOBAY
; -----
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
FTPSEVE X
ENDAUTOLOG
; -----
...
```

5. Save your changes with the **FILE** subcommand:

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

6. Repeat the previous steps on all other members of the SSI cluster.

4.9.2 Reconfigure 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

Follow these 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 MAINT630 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.

Modify 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.

Follow these 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
```
5. These statements will 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.

4.10 Shut down and IPL 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 MAINT630 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 *ITS0ZVM2*:

```
==> at ITS0ZVM2 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 4.5.1, “Modify features and optimize parameter settings” on page 74.

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.

4.10.1 IPL the other SSI members

You must now IPL the other SSI members. Perform the following steps:

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 **N0** 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 **G0**.
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.

4.11 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: ITS0SSIA
SSI Mode: Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: VV155A on 155A
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 ITS0ZVM1 Joined    2015-04-26 17:53:11 2015-04-26 17:53:11
  2 ITS0ZVM2 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
ITS0ZVM1:
VDISK USER  LIMIT IS    2097152 BLK
```

```

ITS0ZVM2:
VDISK USER   LIMIT IS    2097152 BLK
===> ssicmd query vdisk syslim
ITS0ZVM1:
VDISK SYSTEM LIMIT IS INFINITE,          0000000 BLK IN USE

```

```

ITS0ZVM2:
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.

4.12 Enabling basic system automation

Next, enabling basic system automation is described.

4.12.1 Configure 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, perform 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.

4.12.2 Configuring and enabling the programmable operator facility (PROP)

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.

The following steps enable PROP:

1. Log on as MAINT630 if you are not already.
2. Add a minidisk to the operator ID to use for PROP with the command:

```
====> dirmaint for operator amdisk
```

Complete the DirMaint AMDISK panel, as shown in Figure 4-24 on page 111.

```

-----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 4-24 Complete the DirMaint AMDISK panel

3. After you complete the fields as shown, press PF5 to submit. You will 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.

4. 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

```

5. 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

```

6. Modify the PROP routing table:

```
====> xedit prop rtable U
```

Follow these 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 4-12.

Example 4-12 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 4-13. Press Enter to delete the lines.

Example 4-13 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 jumping to the bottom and then moving up four lines:

```
====> bottom
```

```
====> up 4
```

i. Set the case back to MIXED so that XEDIT will retain 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 certain characters at specific columns. You absolutely must keep entries in their correct columns and keep characters in mixed case, as shown in Example 4-14.

Example 4-14 New routing table entries for PROP

```

*-----
* NOTIFY LOGICAL OPERATOR IF LINUX ABENDS
*-----
/OOPS/                8                DMSP0S    LGLOPR
$DISABLED             8                DMSP0S    LGLOPR
$PSW$                 8                DMSP0S    LGLOPR
$psw$                 8                DMSP0S    LGLOPR
$disconnected/        8                DMSP0S    LGLOPR
/HCP$                 8                DMSP0S    LGLOPR
$FAILURE/             8                DMSP0S    LGLOPR
$failed/              8                DMSP0S    LGLOPR
$Failed/              8                DMSP0S    LGLOPR
$No such/             8                DMSP0S    LGLOPR
$ERROR/              8                DMSP0S    LGLOPR
$error/              8                DMSP0S    LGLOPR
$_MACHINE$            8                DMSP0S    LGLOPR
$cannot open/         8                DMSP0S    LGLOPR
*-----
* DON'T WORRY ABOUT ANY OTHER SCIF OUTPUT FROM MONITORED USERS
*-----
                        8
*-----

```

- k. Save the changes and quit XEDIT:

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

7. 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

```

8. Release and detach the OPERATOR 191 and 692 disks:

```

====> release T ( detach
====> release U ( detach

```

9. 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.

```

10. Set OPERATOR as the secondary user for TCPIP, EREP, MAINT, and MAINT630:

```

====> 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 MAINT630 COMMAND ADD 010 SET SECUSER OPERATOR
DVHREQ2289I Your COMMAND request for ... at * has completed; with RC = 0.

```

11. Log off MAINT630 but hold your connection open so that you can immediately log on as OPERATOR:

```
====> logoff hold
```

12. Log on as OPERATOR.

13. 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 4-15.

Example 4-15 PROFILE EXEC for operator and surrogate operator IDs

```
/** OPERATOR/OP1 PROFILE EXEC A    -- MOD 2015-04-09 PWNNOVAK */  
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'
```

14. Save the changes and quit XEDIT:

====> **file**

15. Remain logged on as OPERATOR but launch an additional 3270 session and log on as **LGLOPR**.

16. Position the 3270 sessions for OPERATOR and LGLOPR so that you are able to see both sessions at once.

17. In the session for OPERATOR, issue the command to IPL CMS:

====> **ipl cms**

18. You will see messages in both consoles. The OPERATOR console states that PROP is initializing and then disconnects your session. The LGLOPR console notifies you that PROP initialized with the message:

* MSG FROM OPERATOR: PROP running with routing table PROP RTABLE D1

19. Repeat steps 11 through 15 for every other node in the SSI cluster.

20. 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 absolutely necessary. If you log on as OPERATOR, before you can issue commands other than those commands that control PROP, you must issue the STOP command first.

4.12.3 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 utilized. Simply add one line; an **INCLUDE PROFNAME** statement as the second line of any USER, IDENTITY, SUBCONFIG, or PROTODIR entry that will use that PROFILE.

Consider the following information 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 LNXPWECD

Tip: Avoid directory ambiguity by adhering to a naming standard for your PROFILE entries. Use **LNX**. . . . 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 will create a profile for all Linux virtual machines that is called LNXPDFLT. We will also explain how to review the PROFILE entries supplied by IBM which ship with z/VM, as well as 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.

Create 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.

Follow these steps:

1. Log on as MAINT or MAINT630.
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 4-25. 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 AUTO CR
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 4-25 Contents of the LNXPDFLT PROFILE that is created

The following explanations refer to Figure 4-25:

- 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 lnxpdf1ft
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 could have unexpected and undesired 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 4-26 on page 118.

```

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

File
X E D I T 1

```

Figure 4-26 Contents of IBMDFLT PROFILE entry

6. 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.

```

4.12.4 Creation and use of z/VM User Directory prototypes (PROTODIRs)

You will 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 typical CMS virtual machine, and one for a sample Linux virtual machine.

1. To verify that these are present, issue the following commands as **MAINT630** 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 below.

2. If you do *not* see the response above, then issue the following two commands. Once they have completed, then re-issue the command from step 1 above 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 will use both of these default files.

Create CMSPROTO

DirMaint ships with a basic CMS 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 or MAINT630 on any node in the cluster, perform these steps:

1. Initiate 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
DVHXT1191I 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 4-16 below:

```
====> xedit cmsproto protodir A
```

Example 4-16 Edit the file

```
USER CMSPROTO NOLOG 32M 64M G
  INCLUDE TCPCMSU
  IPL CMS PARM FILEPOOL VMSYSU AUTOGR
  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.
DVHXT1191I 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 an existing z/VM User Directory prototype

Perform the following steps to modify a prototype which you have already created:

1. Request the prototype record. In this case we are using CMSPROTO as an example:

```
====> DIRMAINT SEND CMSPROTO PROTODIR
```

You substitute **1234** with the actual reader file number each time.

- 2.
3. RECEIVE 1234 = = A
4. XEDIT CMSPROTO PROTODIR A
5. DIRMAINT FILE CMSPROTO PROTODIR A
6. 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.

Create 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.

Create SUBPRO-1

While you are logged in as **MAINT** or **MAINT630** on any node in the cluster, perform these 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:

- a. If your SSI cluster has ONLY ONE member, perform this step:

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

- b. If your SSI cluster has TWO members, perform this step:

```
====> save SUBPRO-1 PROTODIR A
====> c/1/2/* * # save SUBPRO-2 PROTODIR A
====> file
```

- c. If your SSI cluster has THREE members, perform this step:

```
====> save SUBPRO-1 PROTODIR A
====> c/1/2/* * # save SUBPRO-2 PROTODIR A
====> c/2/3/* * # save SUBPRO-3 PROTODIR A
====> file
```

- d. If your SSI cluster has FOUR members, perform this step:

```
====> 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.

Create 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, follow these steps:

1. Initiate 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 so that it looks like Example 4-17:

```
====> xedit lnxproto protodir A
```

Example 4-17 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 will be 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 need to change the value to 10016 instead as shown below in Example 4-18. 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 4-18 Contents of LNXPROTO with 10016 cylinder minidisk

```
USER LNXPROTO NOLOG
INCLUDE LNXPDFLT
```

DIRMAINT **AMDISK** is described in section 4.16, “Creating identity LNXADMIN for Linux administration” on page 155 and with greater detail in section 6.4.10, “Add a minidisk to a user or identity” on page 200.

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.
...
```

Note: You will see more than one line for the CMS and LINUX protodirs. *This is expected and is not an error.*

Reviewing contents of an existing z/VM User Directory prototype

To review the contents of a prototype, you must know the name of it. We obtained a full listing of all prototypes in the section above, “Listing of all z/VM User Directory prototypes”.

Using LNXPROTO as an example, the following command will quickly show 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 ...
...
Ready; T=0.01/0.01 13:45:08
DVHREQ2288I Your CMS request for DIRMAINT at * has been accepted.
DVHCMS3868I USER LNXPROTO NOLOG
DVHCMS3868I      INCLUDE LNXPDFT
DVHCMS3868I      AMDISK 0100 3390 AUTOG 5008 POOL1 MR
DVHREQ2289I Your CMS request for DIRMAINT at * has completed; with RC = 0.
```


4.12.5 Create a time-based virtual service machine named CRONSVM

Create a virtual machine that will be 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**.

Follow these steps:

1. Log on as **MAINT** or **MAINT630** on any cluster member if you are not already.
2. Issue 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 ...
```

Change the directory entry so that it looks like Example 4-19:

```
===> xedit cronsvm direct a
```

Example 4-19 Change the directory entry

```
USER CRONSVM LBYONLY 32M 32M ABCDEFG
  INCLUDE TPCMSU
  ACCOUNT 3 OPERATOR
  LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNNOVAK 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: 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.

Tip: If you accidentally delete or modify the *DVHOPT line, use the XEDIT subcommand **QUIT** 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 **QUIT** 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 ...
DVHXT1191I 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.

```

4. Link to the CRONSVM 191 minidisk read/write as file mode X:

```

===> vm link cronsvm 191 < C191 X MR >
DMSVML2060I CRONSVM 191 linked MR as C191 file mode X

```

5. 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 4-20:

```

===> copy VMCRON EXEC M = = X (olddate
===> xedit vmcron exec X

```

Example 4-20 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
  end
end

```

```

else Do
    'DROPBUF'
    leave
end
/* end of else Do */
end
/* end of Do forever loop */
exit

```

6. Create the PROFILE EXEC for CRONSVM by using the information that is in Figure 4-27.

```

/*** CRONSVM PROFILE EXEC A : CRONSVM 191 -- MOD 2015-04-09 PWNNOVAK ***/
'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 4-27 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:

```

HPCRC8083I 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.

4.12.6 Create VMLOGS

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 tarball 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.

Follow these steps:

1. Log on as **MAINT** or **MAINT630** on any cluster member if you are not already logged on.
2. Create the following directory entry by using Example 4-21:

```
===> xedit vmlogs direct a
```

Example 4-21 VMLOGS directory entry

```
USER VMLOGS LBYONLY 64M 128M ABDEG
  INCLUDE IBMDFLT
  IPL CMS PARM AUTOOCR
  LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNNOVAK 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:MAINT630.UTILS.VMARC P (forcerw
```

```
===> access VMPSFS:MAINT630.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 PARS 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/MAINT630, and sets MAINT/MAINT630 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;
HPCPCFX6768I 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: HPCPCFX6769I Your SECUSER terminated by MAINT630.
HPCPCFX6769I 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

```

4.12.7 Shut down and IPL the SSI cluster again

It is recommended that you again shut down and IPL to test the changes:

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 4-22.

```

===> ssicmd query alloc page

```

Example 4-22 Results of SSICMD EXEC

```

ITS0ZVM1:
      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%

ITS0ZVM2:
      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.
4. In 4.10, “Shut down and IPL the SSI cluster again” on page 106, this task was accomplished manually:

```
===> shutdown
```

```
...
```
5. 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.
6. After the system comes back, log on as MAINT.
7. 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 K pages each, or about 6.9 GB of page space. (A page is 4 KB.) In total, you will have 3522 K per member, or about 13 GB of page space. This amount will be sufficient for the setup that is described in this book, but you must monitor the use of pages as an ongoing activity.

4.13 z/VM security and hardening

The following security and system hardening topics are explained:

- ▶ Use an External Security Manager for correct resource security
- ▶ Using LOGONBY for correct accountability
- ▶ Encrypting communications to and from z/VM
- ▶ High-level z/VM security
- ▶ Linux virtual machine privilege classes

4.13.1 Use an External Security Manager for correct resource security

Seriously 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 8.2, “Enable and configure RACF” on page 217.

4.13.2 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 4.12.4, “Creation and use of z/VM User Directory prototypes (PROTODIRs)” on page 118, 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 MAINT630.

While logged on as MAINT or MAINT630, perform the following steps:

1. Issue the DirMaint logonby command to open the LOGONBY panel as shown in Figure 4-28:

```
==> dirmaint for maint630 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:

==> BG

==> FMIRANDA

==> LYDIAP

==> PWNVAK

==> KWERNER

==>

==>

==>

Figure 4-28 DirMaint LOGONBY panel

Press F5 to proceed. You will see the following messages:

```
DVHXT1191I Your LOGONBY request has been sent for processing to DIRMAINT ...
Ready;
DVHREQ2288I Your LOGONBY request for MAINT630 at * has been accepted.
DVHREQ2289I Your LOGONBY request for MAINT630 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 bg fmiranda lydiap pwnovak kwerner
...
```

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 bg fmiranda lydiap pwnovak kwerner
DVHXT1191I 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 command:

```
===> dirmaint for maint logonby ?
DVHXT1191I 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 BG FMIRANDA LYDIAP PWNVAK KWERNER
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 pwnovak 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.

```
DVHXT1191I Your PW request has been sent for processing to DIRMAINT ...
Ready;
DVHREQ2288I Your PW request for PWNVAK at * has been accepted.
DVHREQ2289I Your PW request for PWNVAK 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 4-29 on page 131. 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 / VVVV\  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 maint630 by pwnovak

                                                    RUNNING  ITS0ZVM1

```

Figure 4-29 Log on by using LOGONBY

- The panel will clear and at the next panel that you see is an acknowledgment that CP is going to process this logon for MAINT630 BY PWN0VAK. In Figure 4-30, the password for PWN0VAK is now entered at the ENTER PASSWORD prompt.

```

LOGON MAINT630 BY PWN0VAK
ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):

CP READ  ITS0ZVM1

```

Figure 4-30 LOGONBY password prompt

- After you complete and test the passwords successfully, the passwords for OPERATOR, LGLOPR, MAINT, and MAINT630 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 logon 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 syntax to set one of these special reserved passwords is shown:

```
==> dirmaint for ... setpw lbyonly
```

4.13.3 Encrypting communications to and from 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, clients, 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 **GSKKMAN** 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 **SSLP00L** 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).

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 4-24 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 **INTCLIEN** 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 will 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 4-23, and after the file is edited in Example 4-24.

Example 4-23 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 4-24 TCPIP profile with SSL Server modifications shown

```
; -----
LARGEENVELOPEPOOLSIZE 50 16384
SSLSERVERID * TIMEOUT 30
SSLLIMITS MAXSESSIONS 3000 MAXPERSSSLSERVER 600
; -----
...
; -----
INFORM
  OPERATOR TCPMAINT MAINT MAINT630
ENDINFORM
; -----
PORT
20  TCP FTPSERVE  NOAUTOLOG ; FTP Server
21  TCP FTPSERVE                ; FTP Server
```

```

23  TCP INTCLIEN          ; TELNET Server
...
992  TCP INTCLIEN SECURE ITS0SSIA ; Telnet Server (Secure)
2049 UDP VMNFS            ; NFS Server
2049 TCP VMNFS      NOAUTOLOG ; NFS Server
; -----
INTERNALCLIENTPARMS
SECURECONNECTION REQUIRED
  TLSLABEL ITS0SSIA      ; TLS CERT LABEL OF 8 NUM / UPGRADE 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 MAINT630 are correctly listed as administrators for the VMSYS file pool:

```

===> query enroll admin for all vmsys
Number Of Administrators =      8
MAINT
MAINT630
MIGMAINT
6VMTCP30
VSMGUARD
VSMWORK1
VSMWORK2
VSMWORK3

```

7. Log off TCPMAINT.

8. Log on as **MAINT630**.

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 will be used to create the updated PROFILE TCPIP for the system with 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 4-25. In the prefix area for that line, type **dd** and press Enter. The **dd** will turn red.

Example 4-25 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 4-26:

====> **/Example DTCPARMS Pool**
 ====> **up 8**

Example 4-26 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 4-27. 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 4-27 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 4-28.

Example 4-28 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 with the **FILE** subcommand:

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

13. Copy the new file to the TCPMAINT 198 disk with 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 4-31:

```
====> vmlink tcpmaint 198 < 1198 U MR > (filelist
```

MAINT630 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	2015-04-09	14:19:19
	PROFILE	TCPIORIG	U1	V	73	57	1	2015-04-08	15:45:24

Figure 4-31 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 4-32. If you make a mistake while you are typing, press F2 and start over; *do not use backspace or delete*.

COPY / = DTCPWRKS = (OLDDATE

MAINT630 FILELIST A0 V 169 Trunc=169 Size=5 Line=1 Col=1 Alt=0									
Cmd	Filename	Filetype	Fm	Format	Lrec1	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	2015-04-09	14:19:19
PROFILE	TCPIORIG	U1	V		73	57	1	2015-04-08	15:45:24

Figure 4-32 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 4-33.

MAINT630 FILELIST A0 V 169 Trunc=169 Size=5 Line=1 Col=1 Alt=0									
Cmd	Filename	Filetype	Fm	Format	Lrec1	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	2015-04-09	14:19:19
PROFILE	TCPIORIG	U1	V		73	57	1	2015-04-08	15:45:24

Figure 4-33 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:
- Move your cursor to the **SYSTEM DTCPARMS U1** line and press F11 to open the file in XEDIT.
 - Delete all lines in the current file:


```
====> delete 10
DMSXCG501I 7 line(s) deleted
DMSXSU559W Warning: file is empty
00001 * * * End of File * * *
```
 - 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
```
 - Modify the header so that it reflects the actual file name, purpose, and last person to modify the file as shown in Example 4-29.

Example 4-29 Modified header

```
* =====
* SYSTEM DTCPARMS : TCPMAINT 198 -- MOD 2015-04-18 PWNNOVAK
* Created by: SSLPOOL EXEC -- 18 April 2015 - 18:41:15
* =====
```

- v. Filter by lines that contain an asterisk (*):

```
====> a11/*
```

- vi. Block-delete the lines that contain comments that state that they are examples and where to implement them, as shown in Example 4-30.

Example 4-30 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 ITS0ZVM2 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 ITS0ZVM2 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:

```
====> a11
====> file
```

- g. Press F2 to refresh the display. You see that the record (line) count and date/time stamp differ between SYSTEM DTCPWRLKS 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 MAINT630 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 MAINT630 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 values, then log off as MAINT630:

```
==> erase SSLPOOL DTCTEMP A
==> logoff hold
```

18. Log on as the **GSKADMIN** user ID and allow its default PROFILE EXEC to run. You will 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, then run the **GSKKYMAN** utility. At the top-level menu, select **Create new database** and respond to the prompts as shown in Figure 4-34 on page 142.

```
==> 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 are case-sensitive. File names, paths, and passwords are also case-sensitive.*

20. Select **option 1** to create a key database, and use the default file name of **Database.kdb**. Note the capital 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 4-34 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.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 initialize 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 the heading, “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 using 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 4-35 on page 144 as a guide. Replace the example values with those values that are correct for your environment. The value for the label needs to match the value that was used in step 3 on page 133.

```

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 4-35 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, 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.

36. From this point forward, you need to 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.

4.13.4 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 on the web at the following site:

<http://www.vm.ibm.com/library/zvmsecint.pdf>

For the latest news, pertinent presentations, papers, Redbooks documents, publications, links to press articles and pointers to online discussions, see the z/VM security page:

<http://www.vm.ibm.com/security/>

Click the **Notify Me** link to be automatically notified through email when new information is available.

4.13.5 Linux virtual machine privilege classes

If you are interested in enhancing security by further restricting the z/VM privileges that are granted to each Linux guest, read *Running Linux Guests with less than CP Class G Privilege*, REP-3870.

4.14 Back up and restore 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 V6R3 Installation Guide*, GC24-6246.

Practice restoring a system. You do not want your first restore to be the result of an emergency. After you complete the backup, try to restore your system by following Appendix H, “Restore the z/VM system backup from tape” in the *IBM z/VM V6R3 Installation Guide*, GC24-6246.

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 the following website:

<http://www.vm.ibm.com/related/>

4.15 Create 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.

4.15.1 Characteristics of SFS file pools

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.

4.15.2 Add 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 MAINT630, use XEDIT to create a directory entry for user LNXSERV1:

```
===> xedit LNXSERV1 DIRECT A
```

2. Use the following example directory entry that is shown in Example 4-31 to populate the file. On the planning worksheet, a volume was specifically denoted as the target for this SFS server's minidisks in 2.12.5, "z/VM DASD" on page 36. That volume is VM156A in our example 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 4-31 Sample directory entry for a new SFS server machine

```
USER LNXSERV1 LBYONLY 64M 128M BG
INCLUDE IBMDFLT
  LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNNOVAK 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.

4.15.3 Generate 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 will be named LNX. Follow these steps:

1. Log on to **LNXSERV1** on the first member. CMS will automatically load, but it will display 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 4-36:

```
==> xedit profile exec a
```

```
/** LNXSERV1 PROFILE EXEC : LNXSERV1 191 -- MOD 2015-04-10 PWNNOVAK **/  
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 4-36 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 4-37:

```
===> xedit lnxserv1 dmssparms a
```

```
ADMIN LNXMAINT MAINT MAINT630 FTPSERVE
BACKUP
FILEPOOLID LNX
SAVESEGID CMSFILES
REMOTE
USERS 300
NOSHUTDOWN SIGNAL
NODFSMS
```

Figure 4-37 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 will open and display the contents of the IBM default values for the POOLDEF file, as shown in Figure 4-38 on page 150.

```

$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 4-38 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 4-38 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
DMSWV1120I File LNX POOLDEF A created or replaced
DMSWV1117I FILESERV processing ended at 17:18:11 on 10 Apr 2015

```

9. Back up the file pool control data:

====> **FILESERV BACKUP**

```

DMSWV1117I FILESERV processing begun at 17:47:15 on 10 Apr 2015
DMSWV1121I LNXSERV1 DMSPARMS A1 will be used for FILESERV processing
DMSWV1121I 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
DMSWV1117I 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**

```

DMSWV1117I FILESERV processing begun at 17:47:26 on 10 Apr 2015
DMSWV1121I LNXSERV1 DMSPARMS A1 will be used for FILESERV processing
DMSWV1121I 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**

4.15.4 Add a directory entry for the SFS administration machine

Next, the creation of the LNXMAINT virtual machine is required. LNXMAINT will own and maintain the LNX SFS file pool. Follow these steps:

1. Create the entry and then send it to DirMaint for processing:

```
===> xedit lnxmaint direct a
```

You will see the information that is shown in Figure 4-39.

```
USER LNXMAINT LBYONLY 32M 128M BG
INCLUDE IBMDFLT
IPL CMS PARM FILEPOOL LNX AUTOOCR
  LOGONBY AUTOLOG1 BG FMIRANDA KWERNER PARZIALE PWNNOVAK SPIEDIE
MACHINE ESA
```

Figure 4-39 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 may initially see errors stating 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 pwnnovak
```

```
LOGON LNXMAINT BY PWNNOVAK
```

```
LNXMAINT AT ITS0ZVM1 VIA *          04/10/15 18:37:54 EDT          FRIDAY
```

```
Ready;
```

Example 4-32 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 PWNNOVAK
```

```
RUNNING   ITS0ZVM1
```

3. Enroll LNXMAINT in the LNX file pool with a limit of 500 4K blocks. This amount will be 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 will access 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 4-40 on page 152:

```
==> xedit profile exec a
```

```
/** LNXMAINT PROFILE EXEC : LNX:LNXMAINT. -- MOD 2015-04-10 PWNNOVAK ***/
ADDRESS COMMAND
'CP SET PF11 RETRIEVE BACK'
'CP SET PF12 RETRIEVE'
'EXEC VMLINK .DIR LNX:LNXADMIN. < . D FORCERW >'
'CP SET RUN ON'
EXIT
```

Figure 4-40 LNXMAINT PROFILE EXEC contents

7. Enroll LNXADMIN, which will hold all of the contents that are used to punch the Linux kernel and start the FILESERV GENERATE. We allocate 30,000 blocks to start off, this can be increased later as your environment expands.

```
==> enroll user lnxadmin lnx ( blocks 30000
```

8. Invoke the new profile:

```
==> profile
```

The virtual machine that will be the Linux administrative system is now defined. Remain logged in as LNXMAINT and proceed to perform the initial enrollments.

4.15.5 Enroll the Linux virtual machines as USERS

While you are still logged in as LNXMAINT, perform the initial creation of 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 proper security, the only IDs enrolled in the LNX filepool should be Linux virtual servers, admin IDs defined in Figure 4-37, and I/T staff supporting z/VM and the 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 lnx0001 lnx ( blocks 100
====> enroll user lnx0002 lnx ( blocks 100
====> enroll user lnx0003 lnx ( blocks 100
```

4.15.6 Add Linux parm files and REXX EXECs to the LNX file pool

While you are still logged in as LNXMAINT, perform 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
```

a. 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.)
```

b. 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 4.2.1, "Create directories on the FTP server and upload the installation image" on page 54. We are now ready to transfer files from the FTP server to the LNX file pool using one of the following methods; *either a or b*:

- a. 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

```

- a. Or, perform a **put** (push) from the FTP server by initiating a session to z/VM:

```

===> ftp itsozvm1.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
```

SFS aliases function in a virtually identical way to symbolic links in Unix or Linux. Check aliases 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.

4.16 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 **MAINT630**:

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 4.12.4, "Creation and use of z/VM User Directory prototypes (PROTODIRs)" on page 118:

```
==> dirmaint add LNXADM-1 like SUBPRO-1 build on ITS0ZVM1 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 ITS0ZVM2 in LNXADMIN
DVHXT1191I 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 4-41 on page 156.

```

-----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 ==> VM156Z
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 4-41 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 will not use CMS to format the minidisk.

```
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.

- 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
```

- Assign the **0100** minidisk to **LNXADM-2**:

```
==> dirmaint for LNXADM-1 AMDISK 0100 3390 AUTOV 10016 VM1569 MR PW LNX4VM
LNX4VM LNX4VM
```

- Assign the **0200** minidisk to **LNXADM-2**:

```
==> dirmaint for LNXADM-2 AMDISK 0200 3390 AUTOV 10016 VM156A MR PW LNX4VM
LNX4VM LNX4VM
```

4.17 Monitoring file pool utilization

As any ID that you configured to be an administrator for the LNX file pool, you can review the current utilization 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%

Ready;

====> 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.



Servicing z/VM

“You cannot solve a problem with the same kind of thinking that created it.”

— Albert Einstein

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 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 basically the same.

Awareness is key: Go to the z/VM website and subscribe to notifications for both *Service News* and *Red Alerts*. This topic was covered in 4.3.6, “Service-level validation and subscribing to service notifications” on page 69.

5.1 Typical release schedule

A new release of z/VM is available approximately every 12 - 18 months. In addition to incorporating fixes to previously identified problems, new releases emphasize new function and features that improve virtualization and the use of z/VM as a hypervisor for other z Systems operating systems. Clients are advised to run their production z/VM systems at the currently supported and available version and release.

5.2 Recommended service upgrades (RSUs)

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 prebuilt format. Clients are advised to maintain RSU currency of a minimum of six months on their production z/VM systems.

RSUs (“stacked” or otherwise) are just 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 pre-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, so do not get upset if they 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)
- ▶ Open Systems Adapter Support Facility OSA/SF

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 will be new, but the others will be 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.

Two manuals address the application of corrective service to z/VM:

- ▶ *z/VM Guide for Automated Installation and Service* (see Part 4), GC24-6197:
<http://publibz.boulder.ibm.com/epubs/pdf/hcsk2c00.pdf>
- ▶ *z/VM Service Guide*, GC24-6232:
<http://publib.boulder.ibm.com/epubs/pdf/hcsf1c00.pdf>

These manuals are more complete than this chapter. You might consider using 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).

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. Start at the following website:

<http://www.vm.ibm.com/service/>

The body of the page is similar to the example that is shown in Figure 5-1.

IBM Systems > System z > z/VM >

z/VM Service Resources

This page provides you with a list of VM-related service information and other online services resources.

- [Important VM Service News](#)
- [z/VM Red Alert - Critical Issues](#)
- [Service Tips](#)
- [Using VMSES/E to find RSU Level or PTF/APAR Status](#)
- [Ordering and Installing Service for z/VM and related products](#)
- [RSU, Recommended Service Upgrade](#)
- [ESO: Products serviced by VM ESO with associated current levels](#)
- [z/VM SDO: \(System Delivery Offering\)](#)
- [Daylight Saving Time 2007 Changes \(DST2007\)](#)

Additional Resources

- [Program Directories](#)
- [Preventive Service Planning \(PSP\) Buckets search](#)
- [Announced End-of-Service Dates](#)
- [Technical Support: Technical Information Database \(e.g FLASH \)](#)
- [IBM Global Services - VM and VSE Specialists](#)

IBM Support Portals

- [IBM ShopzSeries](#)

ShopzSeries is an IBM productivity tool for planning and ordering zSeries **software products** and **service**.

- You can order CORrective (APAR/PTF) and preventive (RSU or ESO) service for VM licensed products.
- You can order a variety of tailored software packages for the z/OS, z/OS.e, OS/390, z/VM, VM/ESA and VSE/ESA environments.
- ShopzSeries also lets you review your software licenses in these environments.

Note: Not all ShopzSeries features are supported in all countries.

Service Links

- [IBM System z Library](#)
- [IBM Publications Center](#)
View, download, or order IBM publications.
- [Visit the Adobe Acrobat](#)
Web site for information about downloading the free Adobe Acrobat Reader to read and print PDF files.
- [Looking for Hardware pubs? Try Resource Link](#)
- [Glossaries of IBM Terminology](#)

Figure 5-1 z/VM Service main web page

Consider viewing several of the links on this page.

5.3 How to apply 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).

The following website contains the latest RSU content information:

<http://www.vm.ibm.com/service/rsu/>

The following website contains Red Alerts, which contain information about potential high-impact items:

<http://www.vm.ibm.com/service/redalert/>

Next, we summarize applying service and also describe how to obtain service over the Internet by using IBM Shopz.

You must first determine whether your system needs service. Use the **QUERY CPLEVEL** command:

```
==> query cplevel
z/VM Version 6 Release 3.0, service level 0000 (64-bit)
Generated at 2015-01-01 12:00:00 EDT
IPL at 2015-04-09 17:37:03 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 1403RSU, which is the third RSU for calendar year 2014, and 1501RSU, which is the first RSU of calendar year 2015.

Use the following overall steps in applying an RSU:

1. Get service from the Internet.
2. Download the service files.
3. Receive, apply, and build the service.
4. Put the service into production.

5.3.1 Get service from the Internet

An RSU is obtained by its PTF number. The PTF for the most current RSU is of the form **UM97xyz** where xyz is the z/VM version-release-modification level. So for z/VM 6.2, the RSU is UM97620, and for z/VM 6.3, it is UM97630.

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.

Perform the following steps. (These same steps are described with some screen captures in 5.4, “How to apply a program temporary fix” on page 168.)

1. Point a web browser to the z/VM Service page:

<http://www.vm.ibm.com/service/>

2. Click **IBM Shopz** under IBM Support Portals.
3. Click **Sign In for registered users**, usually in the upper right. Use your user ID and password, if you have them. If not, click **New user registration** and complete the form to create an ID and password. You must have your IBM customer number. (If you work for IBM, note the “IBM employees must sign in here” link.)

4. Click **create new software orders** near the top.
5. The My Orders page opens. Under Package Category, click **z/VM - Service**. Choose **RSU recommended service** in the drop-down menu. Click **Continue**.
6. Five panels of forms are self-explanatory. On panel 3 of 5, choose the radio button that applies to your version of z/VM. In this example, we used z/VM Version 6.2.0 Stacked 6202RSU (PTF UM97620).
7. On panel 4 of 5, choose **Internet** as the delivery mechanism.
8. On panel 5 of 5, complete the form and click **Submit**.
9. In a few minutes, you get two emails. One email is for the core RSU. The other email is for the *PSP bucket* (additional fixes after the RSU). Alternatively, you can click the refresh button on your browser. After a while, Status changes to a link named Download, as shown in Figure 5-2. Click **Download**.

In process orders			
Select	Order reference number - Order name	Status	
<input type="checkbox"/>	U01089290 - Service - 2012-07-11 08.51.35 Customer number: 5471556 IBM order number: B1309082	Download	 History
<input type="checkbox"/>	U01089293 - VM PSP service 2012-07-11 12.54.51 Customer number: 5471556 IBM order number: B1309080	Download	 History

Figure 5-2 Downloading service directly from your browser

5.3.2 Download the service files

In this example, the service files are staged on a desktop machine, then copied to z/VM with FTP. Follow these 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 MAINT630 500 disk. The following example shows an FTP from a DOS session:

```
C:\Downloads>ftp 9.12.7.12
User (9.60.18.249:(none)): maint630
Password:
ftp> cd maint630.500
250 Working directory is MAINT630 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 MAINT630.

4. Access the MAINT630 500 disk as file mode C. Query the disks:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> q 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
6201RSU1 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 time.)

```
==> deterse s1309082 shiprsu1 c = servlink =
```

Usually, this step succeeds. However, large RSUs can fill up the MAINT 500 disk either on the FTP step or the DETERSE step. For example, you might get 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.

5.3.3 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:

```
vmfmrdsd 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. Follow these steps:

1. Log on to a 3270 session as MAINT630.

2. Access the MAINT630 500 disk as C:

```
==> acc 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 parenthesis, 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: MAINT630          ***
*****
***          Date: 07/11/12          Time: 10:54:38          ***
*****
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 EEQINSTMSIBIN
WN:VMFBDC2250W The following OSA objects have been built on BUILD0 100
WN:          (K) and should be copied to your workstation:
WN:VMFBDC2250W IOAJAVA BIN
CK:VMFSRV1233I The following products have been serviced.
CK:VMFSRV1233I CMS CP OSA TCP/IP VMHCD
```

For these example warnings, if you are running OSA or HCD, as the VMFBDC2250W message states, you will need to copy the stated objects to your workstation.

4. Press F3 to get out of **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 V6.2.0 2012-06-26 17:16
...
```

6. Access the MAINT 500 disk again as C:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
```

7. Apply the PSP bucket, if one exists. (In this example, no PSP bucket existed for RSU6302, so 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: MAINT          ****
*****
****          Date: 09/16/10          Time: 15:53:09          ****
*****
R0:VMFAPP2112W PTF UK59536 has a IFREQ requisite for PTF UM33113 in
R0:          product 6VMCMS10 (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 get out of **XEDIT**.

10. Log off from MAINT630.

5.3.4 Put the service into production

We describe how to use the **PUT2PROD** command to put the service into production.

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.

Follow these steps:

1. Log on to MAINT630 on the first member.
2. IPL CMS:

```
==> ip1 cms
z/VM V6.2.0    2012-06-26 17:16
...
```

3. Use the **PUT2PROD** command to put the service into production. Many panels will 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 with 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: LEFT630          USERID: MAINT630          ****
```

```
*****
```

```
****          Date: 07/11/12          Time: 11:16:35          ****
```

```
*****
```

```
CK:VMFP2P1233I The following products have been put into production.
```

```
CK:          Recycle the appropriate servers.
```

```
CK:VMFP2P1233I CMS CP OSA 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 only messages are informational. If warning or error messages exist, you need to address those issues.

5. Press F3 to get out of **XEDIT**.
6. Even though the service was “put into production”, the **QUERY CPLEVEL** command still returns the current service level, which in this example is 1101 (the first RSU in the year 2011) because the new CP load module (nucleus) was not loaded:

```
==> q cplevel
```

```
z/VM Version 6 Release 2.0, service level 1101 (64-bit)
```

```
Generated at 06/27/12 09:00:40 EDT
```

```
IPL at 06/27/12 09:34:06 EDT
```

7. Perform 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 MAINT630.
 - b. Log on to MAINT.
 - c. Issue the **S REIPL** command:

```
==> shutdown
```

```
...
```

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:

```
==> q cplevel  
z/VM Version 6 Release 2.0, service level 1201 (64-bit)  
Generated at 07/11/12 10:55:40 EDT  
IPL at 07/11/12 11:31:09 EDT
```

This information shows that the new CP load module is used and that the service level is the first RSU in the year 2012.

5.4 How to apply 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, VM65060, was opened when a problem was identified with Cooperative Memory Management (CMM).

The APAR was assigned the following PTF numbers for each of the following z/VM releases:

z/VM 5.4	UM33537
z/VM 6.1	UM33538
z/VM 6.2	UM33539

So for z/VM 6.2, you want to apply PTF **UM33539**. The following example shows how to apply this PTF.

5.4.1 Get service by using Shopz

Service for z/VM is still available on tape media. However, getting service over the Internet is more convenient and becoming more common. Perform the following steps:

1. Point a browser to the following URL:
<http://www14.software.ibm.com/webapp/set2/psearch/search?domain=sysz>
2. Enter the APAR number in the Search For: text field. In this example, the APAR is VM65060, and there was one match, as shown in Figure 5-3 on page 169.

System z Technical help
Feedback

Technical help database

for System z

Search

Search for: VM65060

Sort by: Newest first
Hits per doc type: 10

[Search tips](#)
[Start a new search](#)

Document types

☒ All
☒ APARs
☒ FAQs
☒ Product information
☒ Education
☒ Technotes
☒ Flashes
☒ Preventive service planning
☒ Redbook abstracts
☒ Solutions
☒ White papers

[APARs](#)
[Flashes](#)
[FAQs](#)
[Preventive service planning](#)
[Product information](#)
[Redbook abstracts](#)
[Education](#)
[Solutions](#)
[Technotes](#)
[White papers](#)

APARs

1-1 of 1 items found

1. [VM65060: DIAG 10 PERFORMANCE: LINUX CPUPLUGD MEMORY UNPLUG](#)

Performance issue with Diagnose x'10' as used by Linux hot memory unplug function.

2011-11-22

[Back to top](#)

Figure 5-3 Searching for PTFs by APAR number

- Click the link of the APAR description.
- 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 **UM33539** is available for z/VM 6.3. Before you get that PTF, ensure that it was not already applied.

5.4.2 Determine whether a PTF was applied

Check to ensure that the PTF was not previously applied. In this example, we check for the PTF UM33539. Follow these steps:

- Log on to MAINT630.
- Use the **SERVICE ALL STATUS** command followed by the PTF number so that you can query whether it was applied:

```

==> service all status um33539
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE ALL STATUS UM33539
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 ZVM63A

```

```

DASD 01CC LINKED R/W; R/O BY PERSMAP1 at ZVM63A
DASD 029D LINKED R/W; R/O BY      2 USERS
VMFSRV1227I UM33539 is not received or applied
VMFSRV2760I SERVICE processing completed successfully

```

This message shows that PTF UM33539 was *not* applied. Next, we describe how to obtain and apply this PTF.

5.4.3 Download the service to z/VM

From the previous APAR web page search, we clicked the link for **UM33539**, which results in a web page that is similar to Figure 5-4.

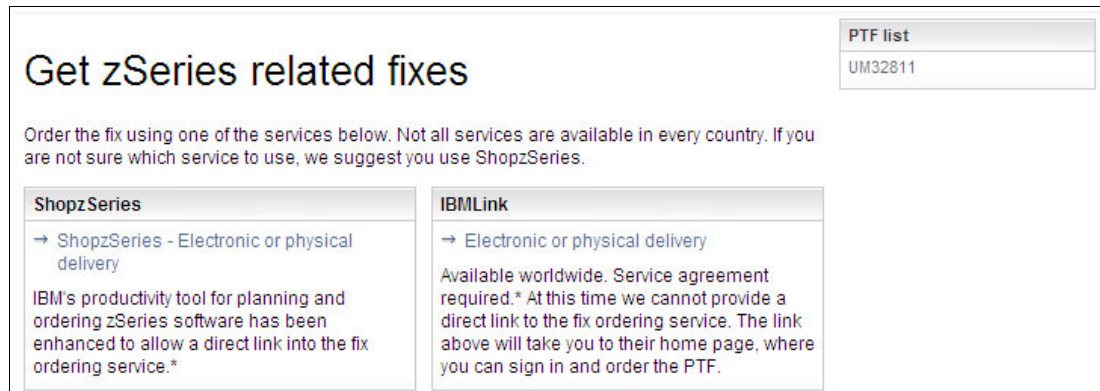


Figure 5-4 Getting fixes from Shopz on IBMLink

Follow these steps:

1. In this example, **ShopzSeries - Electronic or physical delivery** was clicked.
2. Sign in to Shopz with your IBM ID and follow the five self-explanatory steps to order your PTF (if you work for IBM, note the “sign in here” link). When you finish, click **Submit** to place your order.
3. You receive an email within a few minutes. The email shows your order number and a link to start the download of 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

```

4. Point your browser to the link in the email. You see a web page that looks similar to Figure 5-5 on page 171.



Figure 5-5 Web page that was created for downloading a PTF

5. Choose a method of downloading the **VMSES PTF Envelope** and the **VMSES Documentation Envelope** to a desktop or staging machine. In this example, **Download Director** was used.
6. Copy both the SES and the documentation envelopes 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 an FTP from a DOS session:

```
C:\downloads> ftp 9.12.7.12
User (9.60.18.249:(none)): maint630
Password:
...
ftp> cd maint630.500
250 Working directory is MAINT630 500
ftp> bin
200 Representation type is IMAGE.
ftp> quote site fix 1024
200 Site command was accepted.
ftp> mput s1041690.*
150 Storing file 'S1041690.SHIPDOCS'
250 Transfer completed successfully.
ftp: 6144 bytes sent in 0.00Seconds 6144000.00Kbytes/sec.
mput S1041690.SHIPTFSS? y
```

```

150 Storing file 'S1041690.SHIPTFSS'
250 Transfer completed successfully.
ftp: 10240 bytes sent in 0.00Seconds 10240000.00Kbytes/sec.
ftp> quit

```

7. Log on to z/VM as MAINT630.

8. Access the MAINT630 500 disk as C:

```

==> acc 500 c
DMSACC724I 500 replaces C (2CC)

```

9. Verify that the files exist with the **LISTFILE** command:

```

==> listfile * * c
S1041690 SHIPDOCS C1
S1041690 SHIPTFSS C1
6301RSU1 SERVLINK C1

```

10. 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 with 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.

5.4.4 Receive, apply, and build the service

You must receive, apply, and build the PTF. Then, it can be put into production. The process is much easier now with the **SERVICE** command.

To prepare to use the **SERVICE** command, you must have a minidisk with significant free space. Use the MAINT630 500 minidisk for this purpose. Follow these steps:

1. Access the MAINT630 500 disk as file mode C:

```

==> acc 500 c
DMSACC724I 500 replaces C (2CC)

```

2. Use the **SERVICE ALL** command and specify the envelope files that you downloaded. Many panels of output will scroll by and automatically be cleared. Important messages will be saved to the 500 disk. This process can take time. The following example shows the processing:

```

==> service all d1041690
...
VMFSUT2760I VMFSUTFB 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. Invoke 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: MAINT630          ****
*****
****          Date: 01/31/12          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 need to investigate them.

5.4.5 Put the service into production

To put the service into production, perform the following steps:

1. Log on as MAINT630.

2. IPL CMS:

```
==> ip1 cms
z/VM V6.2.0    2011-11-15 11:26
```

3. Access the VMSES/E test build disk as file mode B:

```
==> acc 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 migrations (LGRs) of the important Linux systems in between.
6. Your z/VM system will come back 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 UM33539
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE CP STATUS UM33539
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (6VMCPR20%CP) PTF UM33539 status:
VMFSRV1226I RECEIVED 01/31/12 15:19:15
VMFSRV1226I APPLIED 01/31/12 15:19:15
VMFSRV1226I BUILT 01/31/12 15:19:27
VMFSRV1226I PUT2PROD 01/31/12 15:24:46 POKDEV62
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.

5.4.6 Check for APARMEMO files

After you apply PTFs, check for files with a file type of APARMEMO on the MAIN630T 500 disk. These files might provide instructions for additional work after the PTFs are applied. Perform the following steps:

1. Access the MAINT 500 disk as C and list the files with file type APARMEMO:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * aparmemo c
6VMCPR20 APARMEMO C1
```

This example shows one APARMEMO file.

2. Look at the contents of the file:

```
==> type 6vmcpr20 aparmemo c

APAR MEMOS      01/30/12.14:16:55
=====
```

THE FOLLOWING MEMOS WERE INCLUDED WITH THE PTFS SHIPPED:

NONE.

In this example, the APARMEMO file was created, but no additional memos are present.

You will 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.

5.5 How to determine the service level of TCP/IP

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 the following website:

<http://www.vm.ibm.com/virtualnetwork>

Perform 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:

```
==> q vmlan
q vmlan
VMLAN maintenance level:
  Latest Service: Base
VMLAN MAC address assignment:
  System MAC Protection: OFF
  MACADDR Prefix: 02000B USER Prefix: 020000
  MACIDRANGE SYSTEM: 000001-FFFFFF
  USER: 000000-000000
VMLAN Unified Resource Manager status:
  Hypervisor Access: YES      Status: DISABLED BY SMAPI
  ID: NONE
  MAC Prefix: 02D737
VMLAN default accounting status:
  SYSTEM Accounting: OFF      USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 3
  TRANSIENT Limit: INFINITE    Current: 0
```

The **Latest Service**: line shows that no APAR was applied.

2. 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:

```
==> q vswitch vsw1
VSWITCH SYSTEM VSW1      Type: QDIO      Connected: 2      Maxconn: INFINITE
  PERSISTENT RESTRICTED   NONROUTER      Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0B-00-00-01      MAC Protection: OFF
  State: Ready
  IPTimeout: 5      QueueStorage: 8
  Isolation Status: OFF
Uplink Port:
  RDEV: 4203.P00 VDEV: 0600 Controller: DTCVSW1
  EQID: OSA1SET1
  RDEV: 4300.P00 VDEV: 0603 Controller: DTCVSW2 BACKUP
  EQID: OSA1SET1
```

This query shows that the controller is named DTCVSW1.

3. Use the **NETSTAT** command with the controller name to determine the maintenance of the TCPIP MODULE:

```
==> netstat tcp dtcvsw1 level
```

```
VM TCP/IP Netstat Level 630      TCP/IP Server Name: DTCVSW1
```

```
IBM 2818; z/VM Version 6 Release 2.0, service level 1101 (64-bit), VM TCP/IP  
Lev
```

```
e1 630; RSU 0000 running TCPIP MODULE E2 dated 09/30/11 at 06:55
```

```
TCP/IP Module Load Address: 00C15000
```

This command shows information about the current TCPIP MODULE.

4. Use the **TCPSLVL** command and the complete file specification (TCPIP MODULE E in this example) to get more information. The latest APAR that was applied to TCT00SD is significant.

```
==> tcpslvl tcpip module e
```

```
DTCLVL3306I SLVL data obtained; file TCPIP SLVLDATA A created
```

```
==> x TCPIP SLVLDATA
```

```
...
```

```
SLVL TCPIP      ZVM630
```

```
...
```

```
SLVL SLVL TCT00SD  ZVM630
```

```
...
```

5.6 Moving on to Linux

The installation, configuration, and service of z/VM are complete. z/VM requires little maintenance. It is now time to change your focus to Linux.



Planning and preparing for Linux workloads

“Example isn’t another way to teach. It is the only way to teach.”

— Albert Einstein

This chapter describes the necessary steps to begin your first Linux installation. It describes common tasks that are executed during administration, maintenance, and expansion to accommodate additional workloads.

6.1 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 covers procedures to implement or configure each type. The Linux virtual machine definitions are out of scope of this chapter.

6.2 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.

6.2.1 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 6-1 on page 179 shows the standard 3390 DASD models.

Table 6-1 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 machine systems

When you use DASD, it is helpful to track all of the different DASD numbers and assignments by using the planning worksheet in Appendix A, “References, cheat sheets, and blank worksheets” on page 307.

Although use cases exist for directly dedicating a 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 following **DIRMAINT** command, as shown in Example 6-1. DirMaint will automatically place the entry at the correct location in the directory entry for you.

Example 6-1 Dedicate DASD number 1570 with virtual address 0100

```
====> dirmaint foruser linux2 dedicate 0100 1570
DVHXMT1191I Your DEDICATE request has been sent for processing to
DVHXMT1191I DIRMAINT at ITS0ZVM1.
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 0100
```

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 then possible to use a real volume's alias devices to run more than one virtual machine I/O at a time to the volume. This capability helps to prevent I/O operations from blocking one another. This capability usually results in virtual machine I/O operations with reduced response time, which in turn allows the workload on virtual machines to run more quickly. HyperPAV is described in further detail in 6.2.5, “HyperPAV” on page 190.

6.2.2 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.

Characteristics of direct-attached Fibre Channel

FCP has specific properties:

- ▶ No disks are dedicated, only adapters. The Linux operating system operates the Fibre Channel adapters, as shown in Figure 6-1.
- ▶ Each adapter can provide a large number of 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 z Systems, any supported FCP adapter, such as FICON Express or FICON Express2, can be used for this purpose.

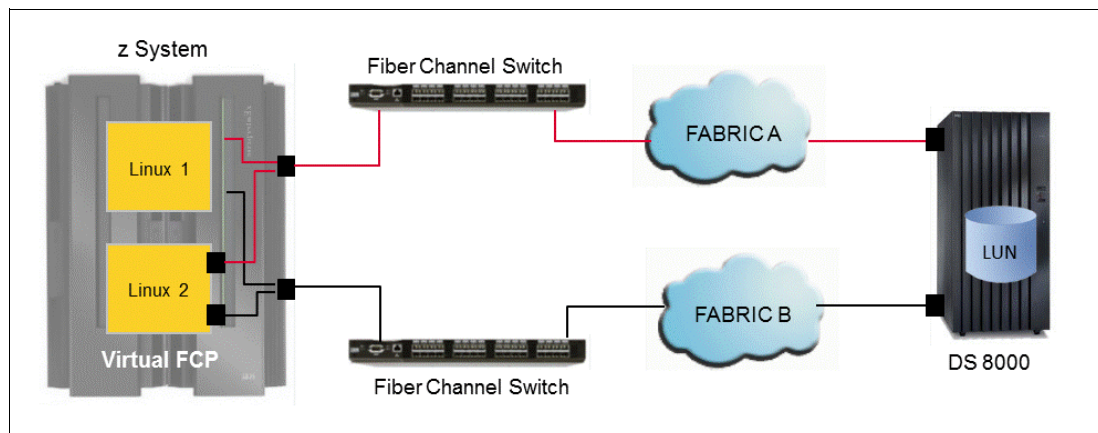


Figure 6-1 Connection overview between disk storage subsystem and virtual machines

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.

Configure 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, use the **SET RDEVICE** command. Execute it on each single system image (SSI) LPAR where Linux needs to be able to relocate. Follow these 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 the steps of this procedure on the other nodes of the SSI.

To make the EQIDs permanent, perform 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 6-2 on page 183.

Example 6-2 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 B8xx dedicated as FC00, and B9xx 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.

6.2.3 Emulated DASD

To simplify the handling of Fibre Channel adapters, you can also define emulated DASD (EDEV), as shown in Figure 6-2 on page 184. EDEVs are based on Fibre Channel, but they still show up as DASD on z/VM.

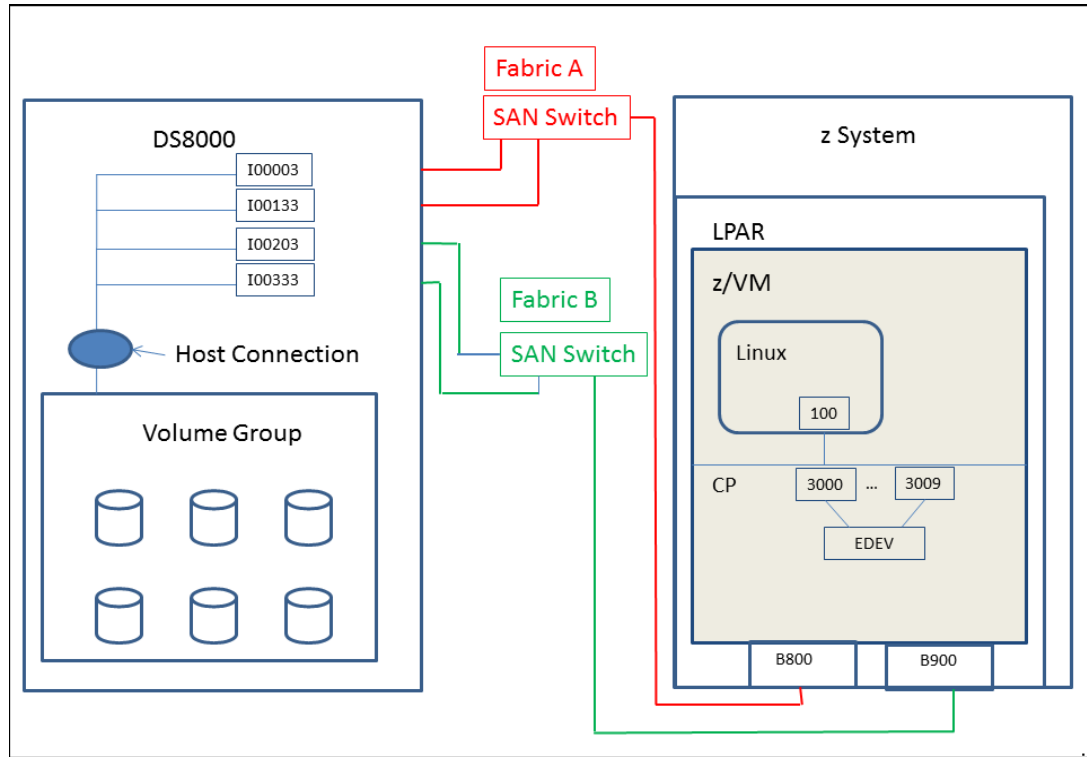


Figure 6-2 EDEV setup overview

Figure 6-2 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 z System 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.

Characteristics of emulated DASD

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.

Configure 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 will be 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 these 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 6-3.

Example 6-3 EDEV SYSTEM CONFIG

```
ITS0ZVM1: begin
  EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107,
    FCP_DEV B800 WWPB 500507630500C74C LUN 4010401100000000,
    FCP_DEV B800 WWPB 50050763050BC74C LUN 4010401100000000,
    FCP_DEV B900 WWPB 500507630510C74C LUN 4010401100000000,
    FCP_DEV B900 WWPB 50050763051BC74C LUN 4010401100000000
ITS0ZVM1: end
ITS0ZVM2: begin
  EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107,
    FCP_DEV B800 WWPB 500507630500C74C LUN 4010401100000000,
    FCP_DEV B800 WWPB 50050763050BC74C LUN 4010401100000000,
    FCP_DEV B900 WWPB 500507630510C74C LUN 4010401100000000,
    FCP_DEV B900 WWPB 50050763051BC74C LUN 4010401100000000
ITS0ZVM2: end
```

5. Save and exit XEDIT with **FILE**.
6. Run these 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 this command:
====> **RELEASE Z (DETACH**

To perform the online configuration without restarting z/VM, follow these steps:

1. Create a small REXX script that enables the EDEV:
====> **XEDIT SETEDEV EXEC A**
2. Insert the lines that are shown in Example 6-4.

Example 6-4 Online definition of EDEV devices with SETEDEV EXEC

```
00001 /* REXX */
00002 'SET EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107',
00003   'FCP_DEV B800 WWPB 500507630500C74C LUN 4010401100000000',
00004   'FCP_DEV B800 WWPB 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 6-5.

Example 6-5 Insert these lines

```
/* REXX */
'SET EDEV 3000 EQID EDASD0 TYPE FBA ATTR 2107 ADD PATH',
'FCP_DEV B900 WWPB 500507630510C74C LUN 4010401100000000',
'FCP_DEV B900 WWPB 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 Example 6-6.

Example 6-6 Output of QUERY EDEV 3000 DETAILS

```
QUERY EDEV 3000 DETAILS
EDEV 3000 TYPE FBA ATTRIBUTES 2107
  VENDOR: IBM PRODUCT: 2107900 REVISION: .194
  BLOCKSIZE: 512 NUMBER OF BLOCKS: 20971520
  PATHS:
    FCP_DEV: B800 WWPN: 500507630500C74C LUN: 4010401100000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: B800 WWPN: 50050763050BC74C LUN: 4010401100000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: B900 WWPN: 500507630510C74C LUN: 4010401100000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
    FCP_DEV: B900 WWPN: 50050763051BC74C LUN: 4010401100000000
      CONNECTION TYPE: SWITCHED STATUS: ONLINE
  EQID: 6005076305FFC74C00000000000010F1F100000000013FFFFF
```

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

Execute these 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. The details and download instructions are at this website:

<http://www.vm.ibm.com/download/packages/descript.cgi?edevpath>

6.2.4 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.

Characteristics of minidisks

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 (valid). 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.

Configure 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 of 4.7.4, “Update the SYSTEM CONFIG file” on page 89.

Define 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 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 will be 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  DEVTPE  -----  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
```

The following explanations refer to 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 0100 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.

6.2.5 HyperPAV

HyperPAV is not a disk device. It is a special device that allows CKD DASD and CKD minidisks to execute 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.

Characteristics of HyperPAV

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 same full-pack minidisk by using multiple LINKs.

- ▶ HyperPAV minidisks with operating system exploitation

An exploiting operating system in a virtual machine is able to 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.

Configure z/VM to use HyperPAV

Example 6-7 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 6-7 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.
DVHDC3451I 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.

Use 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 0600 is created; devices 0600-0602 defined
00: NIC 0600 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 6.4 or use the **dasd_configure** command in SLES) as described in 10.1, “Add disk space to virtual machines” on page 270.

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 will need 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** will overwrite 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 than 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 result will be the same. Twenty 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
```

You will achieve the same 20 aliases for five devices with the following definition, but 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.

6.3 Network attachment options and considerations

Most servers need network connectivity. On z Systems, different networking methods are available, which are provided by either the I/O subsystem that controls the LPARs, such as IBM PR/SM™, or by the z/VM hypervisor. Several additional networking methods exist that are not described in this book due to impracticality, complexity, or limitations. If you have a new installation, we recommend that you use one of the options that are described in this section.

6.3.1 VSWITCH interfaces

The VSWITCH type of network is provided to z/VM virtual machine 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 virtual machine operating system. Under virtually all circumstances, z/VM VSWITCHES must be configured as ETHERNET/Layer 2.

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 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.
- ▶ Both access and trunk ports can be configured for VSWITCHES.
- ▶ VSWITCH network interfaces always operate on port 0 of the virtual device.

Configure z/VM to use VSWITCH interfaces

No special action must be taken to use the VSWITCH interfaces that are at the address triple 600, 601, and 602.

6.3.2 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 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.

Characteristics of direct-attached OSA network devices

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

- ▶ If a dual-port OSA is attached directly to a virtual machine, the virtual machine 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 one time, which is 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.

Configure 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 appropriate OSA device addresses to the virtual machine.

The following rules apply to configuring z/VM to provide direct-attached OSA interfaces:

- ▶ 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.

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

- ▶ The first device gets 0600, 0601, and 0602.
- ▶ The second device gets 0604, 0605, and 0603.

Example 6-8 shows a sample entry in the user directory for the triple 2104, 2105, and 2103 that maps to 600, 601, and 602.

Example 6-8 Direct-attached OSA

```
USER LINUX3 LNX4VM 768M 1G G
...
DEDICATE 0600 2104
DEDICATE 0601 2105
DEDICATE 0602 2103
```

6.3.3 IBM HiperSockets (HIPERS)

IBM HiperSockets™ is a networking option that is defined in the I/O configuration of the mainframe. This type of network connection provides only an internal connection within one Channel Subsystem (CSS).

The main usage for this type of network option is as an internal connection between two z Systems LPARs on the same CPC. An example is the connectivity between a z/OS LPAR that serves IBM DB2® and a z/VM LPAR where one or more Linux virtual machines access the DB2 instance on z/OS.

With the advent of 10 Gb Ethernet connectivity, the utility from HIPERS is diminished.

Characteristics of HiperSockets

HiperSockets have the following characteristics:

- ▶ They are used to configure the connection between two z Systems 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).
- ▶ 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, offloading 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.

Configure 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 6-9 shows an entry in the user directory for the triple 5500, 5501, and 5502 that maps to only the same device addresses.

Example 6-9 User directory entry for numbering devices for HiperSockets

```
USER LINUX3 LNX4VM 768M 1G G
...
  DEDICATE 5500 5500
  DEDICATE 5501 5501
  DEDICATE 5502 5502
```

6.4 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 functionality.

You will need to 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.

6.4.1 Characteristics of DirMaint and the user directory

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.

6.4.2 Checking the status of DirMaint and subcomponents

The DirMaint QUERY command opens the query panel (Figure 6-3), 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 this command:

====> **dirmaint query**

-----DirMaint QUERY-----

Request current system information from the DIRMAINT server.

Select or fill in 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 6-3 DirMaint QUERY panel

6.4.3 Add 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 4.12.4, “Creation and use of z/VM User Directory prototypes (PROTODIRs)” on page 118.

Using the LNXPROTO prototype and optionally CLONEDISK

Run this command to use the LNXPROTO prototype. Executing 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 4.15.5, “Enroll the Linux virtual machines as USERS” on page 152.

6.4.4 Add a USER to z/VM without the use of a prototype

Follow these steps:

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 4.15.5, “Enroll the Linux virtual machines as USERS” on page 152.

6.4.5 Add an IDENTITY to z/VM by using a prototype

This process is described in 4.16, “Creating identity LNXADMIN for Linux administration” on page 155.

6.4.6 Add an IDENTITY to z/VM without using 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. Follow these 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 ITS0VM1 in LNXADMIN
===> dirm add LNXADM-2 build on ITS0VM2 in LNXADMIN
```

4. Erase the temporary work files:

```
===> erase * direct a
```

6.4.7 Change 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 logon time. The value for **MAXSTORAGE** defines how much memory can be used by issuing **define storage**. Several examples are shown:

- 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
```

6.4.8 Modify a user

To change a profile, user, or identity definition manually, first get it, then modify it, and afterward upload it again:

1. Get the entry from the directory:

```
===> dirm for LNXPDFLT get
DVHXMT1191I Your GET request has been sent for processing to DIRMAINT at
DVHXMT1191I ITS0ZVM1 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 so you do not need to retain the temporary copy.

```
===> erase LNXPDFLT direct A
```

6.4.9 Delete 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
```

6.4.10 Add a minidisk to a user or identity

The setup procedure for this task is described in “Configure z/VM to use minidisks” on page 188.

6.4.11 Get a copy of the user directory

You can 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
```

6.4.12 Get and update 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. Follow these 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
```

6.4.13 Clean up the work units

Sometimes, the DirMaint commands never finish and leave work units behind. In general, this situation rarely happens. 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, follow these steps:

1. Check for active work units:

```
===> dirm status workunit all
DVHXMT1191I Your STATUS request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1 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 ITS0ZVM1 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
```

6.4.14 Check the 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
```

Afterward, receive the file, or use peek to look at the results.

6.4.15 Dedicate 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   ITS0ZVM2
```

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 command:

```
====> dirm for linux4 crypto ?
DVHXMT1191I Your CRYPTO request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1 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
DVHREQ2289I Your CRYPTO request for LINUX4 at * has
DVHREQ2289I completed; with RC = 0.
```




Part 2

Other topics

This part of the book includes the following chapters:

- ▶ Chapter 7, “z/VM live guest relocation” on page 205
- ▶ Chapter 8, “z/VM Systems Management API (SMAPI) and Resource Access Control Facility for z/VM (RACF/VM)” on page 209
- ▶ Chapter 9, “Monitoring z/VM and Linux” on page 245
- ▶ Chapter 10, “Working with disks” on page 269
- ▶ Chapter 11, “Working with networks” on page 287
- ▶ Chapter 12, “Miscellaneous helpful information” on page 299



z/VM live guest relocation

“If the facts don’t fit the theory, change the facts.”

— Albert Einstein

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.

7.1 LGR considerations

An SSI cluster has two 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 only on 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, ITS0ZVM1.
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 ITS0ZVM2
User LINUX1 is eligible for relocation to ITS0ZVM2
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 ITS0ZVM1 to ITS0ZVM2 started
User LINUX1 has been relocated from ITS0ZVM1 to ITS0ZVM2
```

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.



z/VM Systems Management API (SMAPI) and Resource Access Control Facility for z/VM (RACF/VM)

“Science is a wonderful thing if one does not have to earn one’s living at it.”

— Albert Einstein

This chapter describes how to enable and configure the z/VM Systems Management application programming interface (API) (SMAPI), and Resource Access Control Facility for z/VM (RACF/VM), which is a z/VM External Security Manager (ESM).

If you want to turn on SMAPI, which is required by certain systems management solutions, you must also have a Directory Maintenance product that was configured as a prerequisite. DirMaint is described here, but also CA products, such as *VM:Secure*, are popular.

Certain organizations’ security policies require an ESM. RACF is described here. CA’s VM:Secure is also an ESM.

This chapter assumes that you completed the setup of DirMaint in 4.8, “Enabling the IBM Directory Maintenance Facility (DirMaint)” on page 92, or another directory management product of your choice.

8.1 Configure DirMaint

To configure DirMaint, perform the following steps:

1. Change the passwords of certain virtual machines so that they can be logged on to:
 - a. Log on to MAINT on the first member of the single system image (SSI) cluster.
 - b. Change the passwords of DIRMAINT, DIRMSAT, DIRMSATx (where x is 2, 3, or 4, depending on the number of SSI members), DATAMOVE, and DATAMOVx from AUTOONLY to your chosen password:

```
====> x user direct
====> /user dirmaint
USER DIRMAINT DIRMAINT 128M 256M BDG
IPL CMS PARM AUTOCR
```

- c. Run the **DIRECTXA** command as MAINT on all members to bring the changes online.
2. Log on to 6VMDIR30 on the first member of the SSI cluster.
3. Access the 492 disk as E to get access to the **DIR2PROD EXEC**:

```
====> acc 492 e
```

4. Use the **DIR2PROD EXEC** to access the necessary minidisks:

```
====> dir2prod access_new 6vmdir30 dirm
DMSACP726I 492 E released
DIR2PROD: Normal Termination.
```

5. Three new minidisks exist that are accessed as J, K, and L (highlighted):

```
====> q disk
```

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSZ	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
DRM191	191	A	R/W	9	3390	4096	2	12-01	1608	1620
MNT5E5	5E5	B	R/O	18	3390	4096	133	1666-51	1574	3240
MNT51D	51D	D	R/W	26	3390	4096	270	1474-31	3206	4680
DIR1DF	1DF	J	R/W	12	3390	4096	13	20-01	2140	2160
DRM492	492	K	R/W	15	3390	4096	291	1717-64	983	2700
DRM41F	41F	L	R/W	16	3390	4096	54	689-24	2191	2880
MNT190	190	S	R/O	207	3390	4096	698	22295-60	14965	37260
MNT19E	19E	Y/S	R/O	500	3390	4096	1126	29766-33	60234	90000

6. To access the user directory source statements, link to the MAINT 2CC disk read-only with the **VMLINK** command. The read password will either be the value that you set all passwords to, or if you did not change them, it will be **READ**:

```
====> vmlink maint 2cc
ENTER READ PASSWORD:
DMSVML2060I MAINT 2CC linked as 0120 file mode Z
```

7. Copy the USER DIRECT file from MAINT 2CC (file mode Z) to DIRMAINT 1DF (file mode J) as the file USER INPUT. This command will cause the current user directory to be loaded into DirMaint when it starts for the first time:

```
====> copy user direct z = input j
```


8. Create the main DirMaint configuration file, CONFIGAA DATADVH L. The L disk is on DIRMAINT 41F, which is the pre-production disk. Add the following lines:

```
====> x configaa datadvh l
====> a 10
ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
DISK_CLEANUP= YES
ONLINE= IMMED
RUNMODE= OPERATIONAL
RACF_RDEFINE_VMBATCH_DEFAULTS=
```

Notes:

- ▶ 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.
- ▶ If the privacy of residual data is a concern on your system, use DISK_CLEANUP= YES.
- ▶ The ONLINE= IMMED line sets your changes to be made immediately.
- ▶ The RUNMODE= OPERATIONAL line sets directory changes to be made. This value can be set to **TESTING** and the changes will not be made.

9. The RACF_RDEFINE_VMBATCH_DEFAULTS= line will not create a VMBATCH-specific resource entry. Otherwise, DIRMAINT will create a VMBATCH resource for this user ID with this line as a default. The VMBATCH generic resource class is configured in the RACF section that follows this section. If you are not installing RACF, this line can be omitted.
10. Create the AUTHFOR CONTROL file on the J disk (DIRMAINT 1DF). Add 12 lines so that MAINT and LNXADMIN are authorized to perform DirMaint tasks, and also the SMAPI virtual machines VSMGUARD, VSMWORK1, VSMWORK2, and VSMWORK3:

```
====> x authfor control j
====> a 12
ALL LNXADMIN * 140A ADGHOPS
ALL LNXADMIN * 150A ADGHOPS
ALL MAINT * 140A ADGHOPS
ALL MAINT * 150A ADGHOPS
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
```

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.

Many of the DirMaint configuration files are created. The next important file is the EXConfigure SMAPI.

After DirMaint (or another directory maintenance product) is configured, SMAPI can be enabled and configured. To set up SMAPI, perform the following tasks:

1. Set up basic SMAPI configuration.
2. Disable support for ensembles.
3. Start SMAPI at IPL time.
4. Test SMAPI from the Conversational Monitor System (CMS).
5. Test SMAPI from Linux by using smaclient.

8.1.1 Set up basic SMAPI configuration

The following steps need to be performed 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.RE
MOVE
00002 MAINT ALL
00003 VSMPROXY ALL
" 004 ZVMLXAPP ALL
```

6. Press Enter and the line will be 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.

8.1.2 Disable support for ensembles

Assuming that your system will not be ensemble-managed, virtual machines that relate to ensembles need to be commented out in a certain configuration file. To turn off ensembles, perform the following steps:

1. As MAINT, access the 193 disk as file mode G:

```
==> acc 193 g
```

2. Access the SFS vmsys:vsmwork1.data disk read/write as file mode H:

```
==> acc vmsys:vsmwork1.data h (forcerw
```

3. Copy the DMSSISVR NAMES file from MAINT 193 to the SFS disk:

```
==> copy dmssisvr names g = = h
```

4. Edit the DMSSISVR NAMES file and comment out the last four servers in the file by putting asterisks in the first column of each line:

```
==> x dmssisvr names h
```

```
====> /ensembles
```

```
...
```

```
*** the following machines are only available in ensembles ***
```

```
*****
```

```
* Default Management Network Server
```

```
*:server.VSMREQIM
```

```
*:type.REQUEST
```

```
*:protocol.AF_MGMT
```

```
*:address.INADDR_ANY
```

```
*:port.44446
```

```
* Primary Vswitch Controller
```

```
*:server.DTCENS1
```

```
*:type.VCTRL
```

```
* Backup Vswitch Controller
```

```
*:server.DTCENS2
```

```
*:type.VCTRL
```

```
* Management Guest
```

```
*:server.ZVMLXAPP
```

```
*:type.MG
```

These settings ensure that virtual machines that relate to ensembles (especially DTCENS1 and DTCENS2) do not start automatically when SMAPI is started.

8.1.3 Start SMAPI at IPL time

To start SMAPI at IPL time, add one line to the **PROFILE EXEC** on the AUTOLOG1 191 disk. To accomplish this task, perform the following steps:

1. Link the AUTOLOG1 191 disk read/write and access it as file mode I:

```
==> link autolog1 191 1191 mr
DASD 1192 LINKED R/W;
==> acc 1191 i
```

2. Edit the PROFILE EXEC and add one line to start SMAPI:

```
==> x profile exec i
...
/*****
/* Customer processing can be added here */
/*****
"CP XAUTOLOG TCPIP"          /* Start TCPIP */
"CP SET MDC STOR OM 256M"    /* Limit minidisk cache in CSTOR */
"CP SET MDC XSTORE OM OM"    /* Disable minidisk cache in XSTORE */
"CP SET SIGNAL SHUTDOWN 600" /* Allow guests 10 min to shut down */
"CP XAUTOLOG LNXADMIN"       /* Start the Linux admin machine */
"CP XAUTOLOG VSMGUARD"       /* Start SMAPI */
...

```

3. Repeat the previous steps for all other members in the SSI cluster.

Verify that SMAPI comes up at IPL time

Perform these steps to verify that SMAPI comes up after an IPL:

1. Query the virtual machines that are running with the **SSICMD EXEC** and the **QUERY NAMES** command to query all active virtual machines on all members:

```
==> ssicmd q n
ITS0ZVM1:
DIRMSAT2 - SSI
FTPSEVER - DSC , LNXADMIN - DSC , TCPIP - DSC , DIRMAINT - DSC
DTCVSW2 - DSC , DTCVSW1 - DSC , VMSERV - DSC , VMSERV - DSC
VMSERVU - DSC , VMSERVS - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP - DSC , OPERATOR - DSC , MAINT -L0004
VSM - TCPIP

ITS0ZVM2:
VMSERV - SSI , DIRMAINT - SSI
FTPSEVER - DSC , LNXADMIN - DSC , TCPIP - DSC , DIRMSAT2 - DSC
DTCVSW2 - DSC , DTCVSW1 - DSC , VMSERV - DSC , VMSERVU - DSC
VMSERVS - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP - DSC
OPERATOR - DSC
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
SYSTEM SHUTDOWN STARTED
HCPHU960I System shutdown may be delayed for up to 630 seconds
VMSERV : 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
ITS0ZVM1:
DIRMSAT2 - SSI
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
LNXADMIN - DSC , TCPIP - DSC , DIRMAINT - DSC , DTCVSW2 - DSC
DTCVSW1 - DSC , VMSERVP - DSC , VMSERVER - 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

ITS0ZVM2:
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 , VMSERVER - DSC , VMSERVU - DSC
VMSERVS - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP - DSC
OPERATOR - DSC
VSM - TCPIP

```

SMAPI is now running and configured.

8.1.4 Test SMAPI from the Conversational Monitor System (CMS)

To test SMAPI, a REXX EXEC that is named **CALLSM1** is included with the files that are associated with this book in Appendix B, “Additional material” on page 321. You copied the REXX EXEC **CALLSM1** to the MAINT 191 (A) disk in 4.7.2, “Use the CPFORMAT EXEC” on page 87. If the REXX EXEC **CALLSM1** was not copied, you need to copy it to complete this section.

To test SMAPI, perform 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**:

```

===> callsm1

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=>

```

This output shows that SMAPI is working from CMS.

8.1.5 Test SMAPI from Linux by using **smaclient**

The script **smaclient** is a powerful, open source bash wrapper around SMAPI. It is available on the web, starting at the following site:

<http://download.sinenomine.net/smaclient>

To test SMAPI by using **smaclient**, perform 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 with 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, perform the following steps:
 - Download the script from the previous URL to a workstation.
 - 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 ITS0ZVM1 USING SUBCONFIG LNXADM-1
06130733
    BUILD ON ITS0ZVM2 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.

8.2 Enable and configure RACF

This section assumes that a new RACF database is being created. For migrating an existing RACF database, see the RACF Program Directory on the web at the following site:

<http://www.vm.ibm.com/progdir/6vmrac30.pdf>

This section also assumes that `DirMaint` and `SMAPI` were configured according to the previous two sections in this chapter. To configure RACF on a new z/VM 6.3 system, perform the following steps. The first five steps are performed before RACF is started. Steps 6 and 7 put RACF into production. The last step is performed after RACF is in production:

1. Create the RACF command file.
2. Customize SMF.
3. Delete the `ICHRCX02` exit.
4. Copy the RACF databases.
5. Set up the `AUTOLOG1` and `AUTOLOG2` virtual machines.
6. Enable RACF.

7. Put RACF into production on all members.
8. Configure SMAPI to work with RACF.

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 completely taken over by RACF. Executing commands, such as **set vswitch vsw1 grant <<userid>>**, will not affect anything.

After you activate RACF, if you encounter access problems, it is a preferred practice to look at the operator console. Commonly, the access issue is reported on the operator console.

Important: If you plan to enable RACF, consider these words of wisdom:

1. 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.
2. 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.
3. 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.)
4. 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.
5. 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 FlashCopy the 192 to the next in a series of disks. It is useful to have several pre-packaged skeleton activities in **SMFPROF**.

8.2.1 Create the RACF 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 RPIDIRECT SYSUT1 for later use with **RPIDIRECT**, perform the following steps:

1. Log on to MAINT on the first SSI member.

2. Link the 6VMRAC30 191 disk read/write and access it as file mode F:

```
===> link 6vmrac30 191 1191 mr
===> acc 1191 f
```

3. Link the 6VMRAC30 505 disk read/write and access it as file mode G:

```
===> link 6vmrac30 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
DVHXT1191I Your USER request has been sent for processing to DIRMAINT
DVHXT1191I 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 6VMRAC30 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
P0
KDEV62
```

5. If you are not using DirMaint, copy USER DIRECT:

```
===> copy USER DIRECT C = = F
```

6. Create the RPIDIRECT SYSUT1 file from the user directory with the **RPIDIRECT** 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:

```
===> rpidirect user withpass f
```

If you used the USER DIRECT file, run this command:

```
===> rpidirect user direct f
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
```

```
...
***** 4859 Directory records processed *****
```

```
***** RPIDIRECT SYSUT1 CREATED *****
```

7. Make a copy of the newly created RPIDIRCT SYSUT1 file to have a reference:

```
====> copy rpidirect 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 rpidirect 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) genclmd(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 to protect this function, see the “Protecting Alternate User IDs” section of the *z/VM RACF Security Server Auditor’s Guide*, SC24-6212, on the web at the following site:
<http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSR8C10.pdf>
- ▶ The fourth line activates the classes VMBATCH, VMMDISK, VMCMD, VMLAN, and SURROGAT.

10. Move the file to the 6VMRAC30 191 disk (F) with the following commands:

```
====> copy rpidirect sysut1 a = = f
====> erase rpidirect sysut1 a
```

The modified RPIDIRCT SYSUT1 file is now on the 6VMRAC30 191 disk.

8.2.2 Customize 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, perform 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 the *z/VM RACF Security Server Auditor’s Guide*, SC24-6212, on the web at the following site:

<http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSR8C10.pdf>

The **PROFILE EXEC** is now configured for the RACFSMF virtual machine.

Modify the SMF CONTROL file

To set SEVER YES in the SMF CONTROL file on the RACFVM 191 disk, perform 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 i
```

2. Edit the SMF CONTROL file and change SEVER NO to **SEVER YES**:

```
===> x smf control i
====> pre 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 i = = h
```
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 i = = j (rep)
```
6. Log off from MAINT.

The SMF configuration of RACF is now complete.

8.2.3 Delete the ICHRCX02 exit

Modify the RACF exit that is named ICHRCX02 to not allow alternate users to access resources that can be accessed by the FTP and NFS servers. The following high-level steps are required to modify the exit:

1. Access the correct 6VMRAC30 minidisks.
2. Extract the highest level (latest) of the 6VMRAC30 component build list.
3. Copy the extracted build list to a new file with an incremented file number.
4. Comment out the ICHRCX02 object from the build list.
5. Update the VVT table to include a pointer to the new build list.
6. Use the build list to build the local mod RACFLPA module.
7. Place the built 6VMRAC30 module into production.

Perform the following specific steps:

1. Log on as 6VMRAC30 on the first SSI member.
2. Issue the following **VMFSETUP** command (1) (The numbers in parentheses correspond to the numbers in the list of high-level steps above.):

```
====> vmfsetup 6vmrac30 racf
```

```
VMFSET2760I VMFSETUP processing started for 6VMRAC30 RACF
```

```
VMFUTL2205I Minidisk|Directory Assignments:
```

	String	Mode	Stat	Vdev	Label/Directory
VMFUTL2205I	LOCALSAM	E	R/W	2C2	RAC2C2
VMFUTL2205I	APPLY	F	R/W	2A6	RAC2A6
VMFUTL2205I		G	R/W	2A2	RAC2A2
VMFUTL2205I	DELTA	H	R/W	2D2	RAC2D2
VMFUTL2205I	BUILD0	I	R/W	29E	RAC29E
VMFUTL2205I	BUILD6	J	R/W	599	RAC599
VMFUTL2205I	BUILD4	K	R/W	505	RAC505
VMFUTL2205I	BUILD2	T	R/W	590	RAC590
VMFUTL2205I	BUILD8	U	R/W	651	RAC651
VMFUTL2205I	BASE	V	R/W	2B2	RAC2B2
VMFUTL2205I	-----	A	R/W	191	RAC191
VMFUTL2205I	-----	B	R/O	5E5	MNT5E5
VMFUTL2205I	-----	D	R/W	51D	MNT51D
VMFUTL2205I	-----	S	R/O	190	MNT190
VMFUTL2205I	-----	Y/S	R/O	19E	MNT19E

```
VMFSET2760I VMFSETUP processing completed successfully
```

3. Many RACF disks were accessed with the **QUERY DISK** command:

```
====> q disk
```

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSZ	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
RAC191	191	A	R/W	25	3390	4096	6	155-03	4345	4500
MNT5E5	5E5	B	R/O	18	3390	4096	133	1666-51	1574	3240
MNT51D	51D	D	R/W	26	3390	4096	270	1474-31	3206	4680
RAC2C2	2C2	E	R/W	9	3390	4096	0	7-00	1613	1620
RAC2A6	2A6	F	R/W	9	3390	4096	8	15-01	1605	1620
RAC2A2	2A2	G	R/W	9	3390	4096	2	9-01	1611	1620
RAC2D2	2D2	H	R/W	70	3390	4096	146	10257-81	2343	12600
RAC29E	29E	I	R/W	10	3390	4096	60	307-17	1493	1800
RAC599	599	J	R/W	31	3390	4096	44	2506-45	3074	5580
RAC505	505	K	R/W	41	3390	4096	132	5137-70	2243	7380
MNT190	190	S	R/O	207	3390	4096	698	22295-60	14965	37260
RAC590	590	T	R/W	63	3390	4096	24	3631-32	7709	11340
RAC651	651	U	R/W	1	3390	4096	2	35-19	145	180
RAC2B2	2B2	V	R/W	85	3390	4096	2330	12607-82	2693	15300
MNT19E	19E	Y/S	R/O	500	3390	4096	1126	29766-33	60234	90000

4. Obtain the latest level of the **RPIBLLPA EXEC** with the following **VMFSIM** command (2):

```
====> vmfsim getlvl 6vmrac30 racf tdata :part rpibllpa exc (history)
:PART RPIBLLPA EXC00000 BASE-FILETYPE
```

The output gives the file name and file type of the currently installed **RPIBLLPA EXEC** (**RPIBLLPA EXC00000**). You will see the output **BASE-FILETYPE** in the last field. In **VMSES/E** terminology, this field means that no service was performed to this part by IBM or locally by a system programmer (no entries in the IBM and Local Version Vector Tables).

5. Copy the **RPIBLLPA EXEC** from the 2B2 disk (V as in the previous **Q DISK** output) to the 2C2 (E) disk with an incremented file type. In this example, the exec becomes **EXC00001** if **EXEC0000** was last (3):

```
====> copy rpibllpa exec v = EXCL0001 e
```

6. Edit the newly copied file and comment out the five lines to exclude the **ICHRCX02** member (4):

```
====> x rpibllpa EXCL0001 e
====> /ichrcx02

...
*:OBJNAME. ICHRCX02 LEPARMS RENT REUS LET NCAL XREF DCBS SIZE 100K,80K
*:OPTIONS. CONCAT SYSLIB RACFOBJ
*:PARTID. ICHRCX02 TXT
*:OPTIONS. ENTRY ICHRCX02
*:EOBJNAME.
*
:OBJNAME. ICHSFRO0 LEPARMS RENT REUS LET NCAL XREF DCBS SIZE 100K,80K
...
====> file
```

7. Log the local modification to the **RPIBLLPA EXEC** into the local version vector table with the following **VMFSIM** command (5):

```
====> vmfsim logmod 6vmrac30 vvtlcl e tdata :mod lc10001 :part rpibllpa exc
```

8. The 2C2 disk (E) now contains the **6VMRAC30 VVTLCL** and **RPIBLLPA EXCL0001** files:

```
====> listfile * * e
RPIBLLPA EXCL0001 E2
6VMRAC30 VVTLCL E1
```

9. Type the contents of the 6VMRAC30 VVTLC1 file:

```
====> type 6vmrac30 vvtlc1 e

:PART.RPIBLLPA EXC :MOD.LCL0001
```

10. Generate a new RACFLPA LOADLIB by using the **VMFBLD** command (6):

```
====> vmfbld ppf 6vmrac30 racf rpibllpa (all)

...
VMFLLB2219I Processing object RPIRACEX
VMFBLD1851I (2 of 2) VMFBDLLB completed with return code 0
VMFBLD2180I There are 52 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

Be sure that the success message is issued.

11. Link the RACFVM 305 disk read/write and access it as file mode L. If you did not change the passwords, the link password will be **multiple**:

```
====> link racfvm 305 305 mr
ENTER MULT PASSWORD:
====> acc 305 1
```

Note: If you are removing this exit after RACF is already activated, the disk might be linked read-only because RACFVM already has the R/W link. To solve this situation, run the following command:

```
====> SEND CP RACFVM LINK * 305 305 RR
```

After you execute this command, proceed with steps 11 and 12. To re-enable R/W for RACFVM, issue the command:

```
====> SEND CP RACFVM LINK * 305 305 MR
```

Afterward, go to step 13 and log off.

12. Use the **VMFCOPY** command to copy the files from the RACFVM 505 disk (K) to the production disk (L) (7):

```
====> vmfcopy racflpa * k = = 1 (prodid 6vmrac30%racf replace oldd
```

13. Log off from 6VMRAC30.

The RACF exit ICHRCX02 is now disabled.

8.2.4 Copy the RACF databases

In an SSI, 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:

- ▶ Copy the RACFVM 200 and 300 minidisks.
- ▶ Change RACFVM to shared disks.
- ▶ Modify the RACMAINT identity.
- ▶ Define the shared disks in the SYSTEM CONFIG file.

Copy the RACFVM 200 and 300 minidisks

To copy the RACFVM 200 and 300 minidisks to the volumes that will be shared, perform the following steps:

1. Log on to the first SSI member as MAINT.

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).

Change RACFVM to 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.

Perform 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 needed to 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 AUTOOCR
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.

Modify 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. Perform 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 6VMRAC30 590 490 MR
LINK 6VMRAC30 505 305 MR
LINK 6VMRAC30 29E 29E RR
LINK 6VMRAC30 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.

Define 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.

Perform 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.

8.2.5 Set 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.

Perform 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
/*****
/*  AUTOLOG1 PROFILE EXEC                               */
/*****
Address Command
"CP XAUTOLOG RACFVM"
"CP LOGOFF"
===> file
```

6. **Perform the steps in this section** 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.

8.2.6 Enable RACF

To enable RACF, perform the following steps:

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 MAINT630 on the first SSI member.

- b. Issue the following **SERVICE** command to enable RACF. This step needs to be performed on only one member. A number of panels will scroll by:

```
===> service racf enable
...
VMFSET2760I VMFSETUP processing completed successfully
VMFSRV1233I The following products have been serviced.
VMFSRV1233I CP RACF
VMFSRV2760I SERVICE processing completed successfully
```

RACF is now enabled on the CF2 disk. This disk is now on the release 1 volume in z/VM 6.3.

2. Shut down the first SSI member:

```
===> shutdown
...
```

RACF is now be enabled. Shut down all members and the SSI.

8.2.7 Put RACF into production on all members

Important: The next paragraph is extremely important. Read it at least twice.

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 will later perform only the **first and last** subsections on the other members:

1. IPL the member and start RACMAINT.
2. Configure the initial RACF database.
3. Enable DirMaint to RACF on the first member.
4. Set the DirMaint use of the reader with RACF on the first member.
5. Put RACF into production.

IPL the member and start RACMAINT

You must IPL each member of the SSI and start RACMAINT. Perform 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”.
3. The STAND ALONE PROGRAM LOADER (SAPL) window as shown in Figure 8-1 on page 232 opens on the Integrated 3270 Console.
4. Change the *Device Number* to the device number of **Release Volume 1**, not the “Res volume” that normally IPLs. In this example, it was 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 6 RELEASE 3.0

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 8-1 STAND ALONE PROGRAM LOADER window

5. 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
...

```

6. Continue to IPL the member. When the IPL process completes, you will be logged on as OPERATOR. Start the virtual machine RACMAINT. You will 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.

If you completed the next three sections on the first SSI member, proceed to “Put RACF into production” on page 236.

Configure the initial RACF database

The following set of steps needs to 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 will see a message that the password expired. Reset the password by typing the new password twice. Separate the passwords with a forward slash (/). You will 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 6VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 6VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 6VMRAC30.191 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 6 Release 2.0, Service Level 1101 (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 06/22/12
z/VM V6.2.0 2012-06-21 16:54
...
```

4. Set the F12 function key to the command **RETRIEVE**:

```
==> set pf12 ret
```

5. Link and access 6VMRAC30's 505, 191, and 29E disks. Disregard any error messages:

```
==> link 6vmrac30 505 505 rr
RPIMGR031E RESOURCE 6VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 0505 LINKED R/O; R/W BY RACMAINT
==> acc 505 c
DMSACP723I C (505) R/O
==> link 6vmrac30 191 192 rr
RPIMGR031E RESOURCE 6VMRAC30.191 SPECIFIED BY LINK COMMAND NOT FOUND
==> acc 192 b
DMSACP723I B (192) R/O
DMSACP725I 192 also = D disk
==> link 6vmrac30 29e 29e rr
RPIMGR031E RESOURCE 6VMRAC30.29E 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 **RPIDIRECT SYSUT1** file that was created earlier and copied to the 6VMRAC30 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 rpidirect
Processing batch file RPIDIRECT 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 MAINT630 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.

Enable DirMaint to RACF on the first member

Perform the following steps to enable DirMaint to run to RACF:

1. Log on to MAINT. You are asked to change the password.
2. Link to the 6VMDIR30 2C2 disk read-only, which has a sample CONFIGRC DATADVH file:

```
===> vm1link 6VMDIR30 2c2  
DMSVML2060I 6VMDIR30 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.  
DVHPR02008I 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 RACF.

Set the DirMaint use of the reader with RACF on the first member

Because the VMBATCH definitions were deleted in 8.2.1, "Create the RACF command file" on page 218, RACF will give errors when DirMaint sends files to the reader. To address this issue, the CP **TRANSFER** and **TAG** commands need to 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, perform 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.

Put 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 MAINT630 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 MAINT630.

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 HCPRP1036E 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 11/18/2010
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 6.2.0
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 HCPRP1035I 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:

- ▶ “IPL the member and start RACMAINT” on page 231.
- ▶ “Put RACF into production” on page 236.

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.

8.2.8 Configure SMAPI to work with RACF

Perform 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. It is 186 days in this example:

```
===> rac setropts password(interval(186))
```

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 RACLSTED 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 RACLST(FACILITY)
```

Exempt 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 go back and try 8.1.4, “Test SMAPI from the Conversational Monitor System (CMS)” on page 215 and 8.1.5, “Test SMAPI from Linux by using smaclient” on page 216.

8.3 Verify that DirMaint and RACF work together

To add new virtual machines, you need to use DirMaint and RACF commands. Perform 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
DVHXTM1191I Your FILE request has been sent for processing to DIRMAINT
DVHXTM1191I 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
DVHXTM1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXTM1191I 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 new virtual machines.

8.3.1 Configure 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 4.13.2, "Using LOGONBY for correct accountability" on page 129.

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 ITSZVM1.
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 genclmd(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 logon, as shown in Figure 8-2 on page 243.


```

z/VM ONLINE

      / VV      VVV MM      MM
     / VV      VVV MMM     MMM
    / VV      VVV MMMM    MMMM
   / VV      VVV  MM MM  MM MM
  / VV      VVV  MM  MMM  MM
 / VVVVV      MM  M   MM
/ VVV      MM      MM
ZZZZZZ /      V      MM      MM

      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 sysadmin by userid1

ITS0ZVM1                                     RUNNING

```

Figure 8-2 Test the logon

8. You will be asked to change the password at the first logon:

```
logon sysadmin by userid1
```

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 6 Release 3.0, Service Level 0000 (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 V6.3.0 2013-06-04 12:50
```

You can issue a **QUERY USERID** command to see that you are logged on as SYSADMIN with its privileges.

8.3.2 Use 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. Follow these 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.



Monitoring z/VM and Linux

“Not everything that can be counted counts, and not everything that counts can be counted.”

— Albert Einstein

This chapter briefly describes how to monitor z/VM and Linux. For another source about z/VM performance and monitoring, see Chapter 11, “Monitoring performance and capacity”, in the manual *Getting Started with Linux on System z*, SC24-6096, on the web at the following website:

<http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSX0C20.pdf?DT=20130528134905&XKS=hcsh2ac2>

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 is briefly described in this chapter.

9.1 Use basic z/VM commands

z/VM has many commands to monitor the state of the system. **CP INDICATE** is the most commonly used, and other commands are addressed. For more information, see the *z/VM Performance Resources* web page at the following site:

<http://www.vm.ibm.com/perf>

9.1.1 Use the INDICATE command

z/VM has 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.

Conversely, 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 additional 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 will require 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 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.

9.1.2 Use other basic commands

Other useful basic commands are briefly mentioned. All examples are shown from the MAINT virtual machine. The results will differ for users with fewer privileges.

Getting help

To get help on the system, use the **HELP** command. Sometimes, it is hard to find help for the exact command that you are looking for. 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 , DTCMAPI - DSC , PERMAPI - DSC , VSMWORK3 - DSC
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
TCPIP    - DSC , DIRMAINT - DSC , DTCVSW2 - DSC , DTCVSW1 - DSC
VMSERV   - DSC , VMSERV   - 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).

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 6 Release 3.0, service level 1301 (64-bit)
Generated at 06/28/13 14:58:28 EDT
IPL at 09/04/13 10:48:34 EDT
```

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
```

VOLID	RDEV	EXTENT START	EXTENT END	TOTAL	IN USE	HIGH	USED	% ALLOCATION 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. The following list gives the short form of these commands without any of the associated output:

```
====> q da
====> q da free
====> q osa
====> q osa free
====> q v all
```

9.2 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 books:

- ▶ *z/VM Performance Toolkit Guide*, SC24-6156, and *z/VM Performance Toolkit Reference*, SC24-6157, on the web at the z/VM 6.3 bookshelf:
<http://publib.boulder.ibm.com/cgi-bin/bookmgr/XKS/hcsh2ac2>
Search for Toolkit on that page.
- ▶ *The Program Directory for Performance Toolkit for VM*, GI10-0785:
<http://www.vm.ibm.com/progdir/6vmptk30.pdf>
- ▶ *Linux on IBM zSeries and S/390: Performance Toolkit for VM*, SG24-6059:
<http://www.redbooks.ibm.com/abstracts/sg246059.html>

How to set up and use the IBM Performance Toolkit are described briefly:

- ▶ Configure the IBM Performance Toolkit for VM
- ▶ Use the IBM Performance Toolkit for VM

9.2.1 Configure the IBM Performance Toolkit for VM

The Performance Toolkit is installed with z/VM. The configuration is described in the Program Directory for Performance Toolkit for VM, which is at the following website:

<http://www.ibm.com/eserver/zseries/zvm/library>

The following summary explains how to turn it on. Again, configure the product only if you ordered it.

1. Query the priced products that are enabled with the **QUERY PRODUCT** command:

```
===> q product
Product State Description
IBMVMSSI Enabled IBM z/VM Single System Image Feature
6VMDIR30 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FACILITY
(Dir
Maint)
6VMPTK30 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
6VMRAC30 Disabled 00/00/00.00:00:00.$BASEDDR RACF Security Server
6VMRSC30 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
```

2. To enable IBM Performance Toolkit for VM, log on as **MAINT630** and enter the following command:

```
===> service perftk enable
VMFSRV2760I SERVICE processing started
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

You will see a few panels of messages scroll by and finally the success messages appear. Performance Toolkit is enabled for the current z/VM session.

3. This process modifies the SYSTEM CONFIG file by appending a line to the end. Verify that this line was added with the following commands:

```
===> vmlink pmaint cf0
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
===> type system config z
... // many screens cleared
PRODUCT PRODID 6VMPTK30 STATE ENABLED DESCRIPTION '06/05/13.15:22:55.MAINT630
PE
RFKIT Minidisk Install and Service'
```

4. The **QUERY PRODUCT** command shows the change:

```
===> q product
Product State Description
IBMVMSSI Enabled IBM z/VM Single System Image Feature
6VMDIR30 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FACILITY
(DirMaint)
6VMPTK30 Enabled 06/05/13.15:22:55.MAINT630 PERFKIT Minidisk Install and
Service
```



```
6VMRAC30 Disabled 00/00/00.00:00:00.$BASEDDR RACF Security Server
6VMRSC30 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
```

The Performance Toolkit is now enabled. You can also verify it by running the **QUERY PRODUCT** command again.

9.2.2 Configure web browser support

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.
```

3. To change TCP/IP dynamically, use the **OBEYFILE** command:

```
===> netstat obey port 80 tcp perfsvm
VM TCP/IP Netstat Level 630      TCP/IP Server Name: TCPIP

OBEY command response is: OK
OBEY return code = 0
```

4. Issue the **NETSTAT CLIENTS** command to verify your configuration. You want to see that the service that is named PERFSVM is a client. PERFSVM will be shown after a few panels of output:

```
===> netstat clients
...
Client: PERFSVM                Authorization: {none}
Notes Handled: none
Last Touched:  0:03:23
Vmcfc error count: 0
```

If you are configuring central monitoring in a single system image (SSI) cluster, it is enough to configure the web server on only one of the members. Central monitoring enables one member to monitor the other members of the SSI cluster.

9.2.3 Configure PERFSVM

The PERFSVM virtual machine is the Performance Toolkit service machine. Follow these steps to configure it:

1. Log on to PERFSVM. If you successfully enabled the product, you will enter a Performance Toolkit session and see the following text at the top of the panel:

```
FCX001                      Performance Toolkit for VM                      Autoscroll 12
FCXBAS500I Performance Toolkit for VM FL630
16:14:15 Monitor event started -- recording is activated
16:14:15 Monitor sample started -- recording is activated
```

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 will take you to line 190 where the next eight lines are comments that start with an asterisk (*). Perform the following changes:

- Uncomment the second, fourth, sixth, and eighth lines by changing *C to FC.
- 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 will look like the following lines. Save your changes with the **FILE** subcommand:

```
*   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
*   Following command activates Internet interface with SSL
...
====> file
```

If you are configuring central monitoring in an SSI cluster, enable the four FC commands only on one member, which will serve 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 **ZVM63A** with your system identifier):

```
====> x fconrmt authoriz
```

```
====> a 2
```

```
ZVM63A 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 will look like the following example:

```
ZVM63A PERFSVM  DATA
```

```
ITS0ZVM2 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
ZVM63A PERFSVM z/VM6.3 N FCXC1R01

```

If you are configuring central monitoring in an SSI cluster, specify all other members also. Ensure that each member uses a unique resource name. The first member might be **FCXC1R01**, the second member might be **FCXC1R02**, and so on.

```

ZVM63A PERFSVM z/VM6.3 N FCXC1R01
ITS0ZVM2 PERFSVM z/VM6.3 N FCXC1R02
ZVM63C PERFSVM z/VM6.3 N FCXC1R03
ZVM63D PERFSVM z/VM6.3 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.FCXSYSTEM :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 will use **FCXC1R02** and **FCXC1S02**. The third member will use **FCXC1R03** and **FCXC1S03**. And, the fourth member will use **FCXC1R04** and **FCXC1S04**.

8. Make CP start to collect performance data. Start Performance Toolkit automatically after the IPL:

- a. Log on to AUTOLOG1.

- b. Before you press Enter at the VM READ prompt, type **acc (noprof)** so that the **PROFILE EXEC** is not run:

```

LOGON AUTOLOG1
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 V6.3.0    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 600" /* Allow guests 10 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 AUTOLOG1.

9.2.4 Start the IBM Performance Toolkit for VM

To start the Performance Toolkit, perform the following steps:

1. Log on to the PERFSVM virtual machine.
2. Press Enter and the performance toolkit starts through the PROFILE EXEC:

```

FFCX001           Performance Toolkit for VM           Autoscroll 12
FCXBAS500I Performance Toolkit for VM FL630
12:32:15 FCXAPP530I Connected to *IDENT for resource FCXC1R01
12:32:15 FCXAPP530I Connected to *IDENT for resource FCXC1S01
12:32:15 FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
12:32:15 FCXAPP527I User PERFSVM connected on path 0006
12:32:15 FCXAPC535I Connected to resource FCXC1R01 on path 0005, for S&F-Coll
12:32:15 FCXTCP575I WebServer host IP address is 9.12.7.11:00080
12:32:15 FCXTCP590I WebServer interface activated
12:32:15 Monitor event started -- recording is activated
12:32:15 Monitor sample started -- recording is activated

```

Disconnect from PERFSVM now.

```
Command ==> disc
```

The Performance Toolkit is now configured and running.

9.2.5 Use 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, perform the following steps:

1. Point a browser to your z/VM system, for example:

`http://9.12.7.11`

2. You will see a splash screen, then the Web Server Logon window, as shown in Figure 9-1.

IBM
Performance
Toolkit for VM

Remote Performance Monitoring Session Setup

Web Server Logon

You are connected to the data retrieval interface of the Performance Toolkit for VM on system **ZVM63A**. Data retrieval authorization is based on your VM user identification on that system. Please enter your userid and password

VM UserID: Password:

VM Logon BY UserID: (use this userid for password validation when LOGONBY capability is enabled for the Toolkit server machine)

Desired screen layout:

Max. Data Lines: Line length:

Up to 12 kB of data can be retrieved per selection, including all control information. Output may be truncated if space is not sufficient for all lines.

Figure 9-1 Performance Toolkit Web Server Logon window

3. Enter any valid user ID and password and click **Submit**. In this example, PERFSVM was used.
4. The Central Monitoring System Load Overview appears with your system identifiers (*Node-ID*) on the left side.
5. Click your system identifier and the Initial Performance Data Selection Menu window appears as shown in Figure 9-2 on page 256.
6. From this window, you can drill down into many different types of reports.

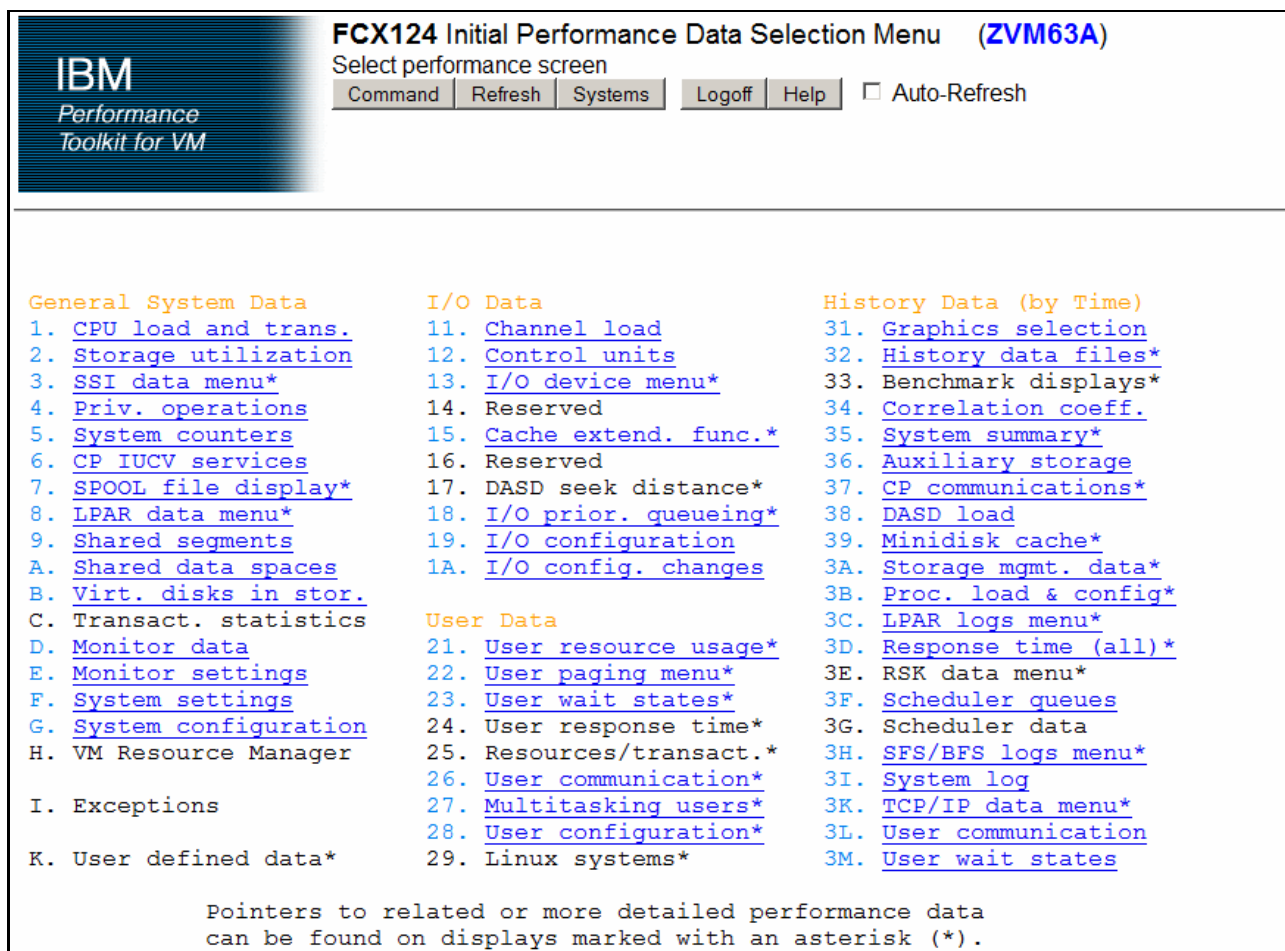


Figure 9-2 Browser interface to the Performance Toolkit

Use a 3270 interface

You can also use a 3270 interface. Perform the following steps:

1. Log on to PERFSVM.
2. If you are disconnected, pressing Enter will get you back 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 panel then appears, as shown in Figure 9-3 on page 257.

FCX124 Performance Screen Selection (FL630) 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. SP00L 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 9-3 Performance Screen Selection panel

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

9.3 Collect and use 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 will compare the current system performance to the past performance so that data will be available for troubleshooting, or to generate reports.

9.3.1 Collect 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 6 Release 3.0, Service Level 0000 (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 V6.3.0    2013-06-04 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 the following site:

<http://www.vm.ibm.com/download/packages/descript.cgi?MONCLEAN>

Follow these 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 will close the output file every hour and execute MONCLEAN EXEC. If the MONCLEAN EXEC was not modified, it will remove the oldest file when the disk reaches 80% full.

8. Example 9-1 shows MONWRITE's 191 disk when **MONCLEAN** is running.

Example 9-1 MONWRITE 191 disk

MAINT FILELIST A0 V 169 Trunc=169 Size=19 Line=1 Col=1 Alt=0									
Cmd	Filename	Filetype	Fm	Format	Lrec1	Records	Blocks	Date	Time
	D061313	T100016	Z1	F	4096	49275	49275	6/13/13	10:29:16
	D061313	T090016	Z1	F	4096	99407	99407	6/13/13	10:00:15
	D061313	T080015	Z1	F	4096	99392	99392	6/13/13	9:00:15
	D061313	T070015	Z1	F	4096	99348	99348	6/13/13	8:00:15
	D061313	T060015	Z1	F	4096	99348	99348	6/13/13	7:00:15
	D061313	T050016	Z1	F	4096	99348	99348	6/13/13	6:00:15
	D061313	T040016	Z1	F	4096	99348	99348	6/13/13	5:00:15
	D061313	T030015	Z1	F	4096	99348	99348	6/13/13	4:00:15
	D061313	T020016	Z1	F	4096	99348	99348	6/13/13	3:00:15
	D061313	T010015	Z1	F	4096	99348	99348	6/13/13	2:00:15
	D061313	T000015	Z1	F	4096	99348	99348	6/13/13	1:00:15
	D061213	T230015	Z1	F	4096	99348	99348	6/13/13	0:00:15
	D061213	T220015	Z1	F	4096	99356	99356	6/12/13	23:00:15
	D061213	T210015	Z1	F	4096	99357	99357	6/12/13	22:00:15
	D061213	T200015	Z1	F	4096	99348	99348	6/12/13	21:00:15
	PROFILE	EXEC	Z1	V	65	2	1	6/12/13	11:35:49
	MONCLEAN	EXEC	Z1	V	75	194	2	6/12/13	11:32:13
	MONCLEAN	README	Z1	F	80	28	1	6/12/13	11:32:13
	MONCLEAN	VMARC	Z1	F	80	64	2	6/12/13	11:32:13

9.3.2 Use 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.

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  
          D061013  T084824  X1 F      4096    53930    53930  6/10/13  9:20:43  
          PROFILE  EXEC      X1 V       65         3         1  6/10/13  8:48:21
```

4. Run the **MONSCAN** subcommand:

```
===> perfkit monscan D061013 T084824 X
```

The regular Performance Screen Selection window opens (Figure 9-4 on page 261).

FCX124	Performance Screen Selection (FL630)) 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 9-4 Performance Screen Selection window

5. Make a selection, for example, 1 - CPU load. The first window will not contain any data. Enter the command **nexts** (for next sample) and a window with real numbers will open. You can see the interval on the top of the window, as shown in Figure 9-5 on page 262.

FCX100		Data for 2013/06/10 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 %		LNADMIN .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		LNADMIN 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 9-5 CPU load performance

9.4 Monitor 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.

9.4.1 Monitor 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:

1. The APPLMON option must be set in the user directory.
2. 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 6.4 and SUSE Linux Enterprise Server (SLES) 11 service pack (SP) 3.

A quick description of how to use this built-in monitoring function is provided:

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 with the `modprobe` command and verify that they were loaded:

```
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

4. Now, 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
```

5. The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory, five files exist:

<code>timer</code>	Controls whether any data gathering is in effect
<code>interval</code>	Sets the interval, in milliseconds, during which samples are taken
<code>mem</code>	Controls the memory data gathering module
<code>os</code>	Controls the CPU data gathering module
<code>net_sum</code>	Controls the net data gathering module

6. 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.

View 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 9-6.

FCX242	CPU 2817	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 9-6 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 9-7.

FCX244	CPU 2817	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 9-7 Linux guest systems memory overview

You can also use a web interface to view the same data.

9.4.2 Monitor Linux performance from inside Linux

Many tools are available for Linux performance monitoring. Several commonly used tools are described. These tools are all platform independent, and they work on Linux in general.

top command

When you run the **top** command without any parameters, it shows a system overview and running tasks, similar to Figure 9-8 on page 265. The output is refreshed every 3 seconds automatically. To leave **top**, press Q.

```

top - 17:26:52 up 7:30, 1 user, load average: 0.06, 0.08, 0.05
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1015640 total, 666560 free, 104228 used, 244852 buff/cache
KiB Swap: 1828452 total, 1828452 free, 0 used. 863024 avail Mem

  PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   9564    6800   4016 S   0.0   0.7   0:00.81 systemd
    2 root        20   0     0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root        20   0     0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
    5 root         0 -20     0      0      0 S   0.0   0.0   0:00.00
kworker/0:0H
    7 root        rt    0     0      0      0 S   0.0   0.0   0:00.00 migration/0
...

```

Figure 9-8 top command

vmstat command

Another useful command is the **vmstat** command, which reports the following information:

- Processes
- Memory
- Paging
- Block I/O
- Traps
- CPU activity

When you run the **vmstat** command without any parameters, it shows only one line that summarizes averages since the last IPL, which is not useful. Figure 9-9 shows the **vmstat 5** command output. This **vmstat 5** command shows the first line with averages since the last IPL and then writes a new line every 5 seconds with the current data.

```

# vmstat 5
procs -----memory----- ---swap-- ----io---- -system-- -----cpu-----
 r b  swpd  free  buff  cache   si   so    bi    bo    in  cs us sy id wa st
 0 0    0 666832 4596 239924    0    0     4     1   79 126 0 0 100 0 0
 0 0    0 666768 4596 239956    0    0     0     0  149 248 0 0 100 0 0
 0 0    0 666752 4596 239956    0    0     0     0  149 246 0 0 100 0 0
 0 0    0 666720 4596 239956    0    0     0     0  171 266 0 0 100 0 0
 0 0    0 666752 4596 239956    0    0     0     0  169 258 0 0 100 0 0
 0 0    0 666752 4596 239956    0    0     0     0  166 254 0 0 100 0 0
 0 0    0 666752 4596 239956    0    0     0     0  160 247 0 0 100 0 0
 0 0    0 666720 4596 239956    0    0     0     0  152 250 0 0 100 0 0

```

Figure 9-9 vmstat command

The **wa** column shows a wait time and represents a percentage of time while the system waited for I/O. The higher the percentage, the more time tasks waste nonproductively.

The **st** column shows what is known as *stolen time*. It represents the time that the CPU was stolen from a guest by the hypervisor. Many reasons exist for stolen time: CPU contention at the z/VM level, heavy z/VM paging, heavy virtual switch usage, and so on. The higher the number, the more time a guest spends nonproductively.

sysstat package

The following tools are part of the sysstat package, which might not be installed automatically.

Red Hat Enterprise Linux 7.1

For RHEL 7.1, follow these steps:

1. Check whether the sysstat package is installed:

```
# rpm -qa | grep sysstat
sysstat-10.1.5-7.el7.s390x
```

2. If it is not installed, install it with the following command:

```
# yum install sysstat
...
Installed:
  sysstat.s390x 0:10.1.5-7.el7
```

```
Dependency Installed:
  lm_sensors-libs.s390x 0:3.3.4-11.el7
```

SUSE Linux Enterprise Server 12

For SUSE 7.1, enter this command:

```
# zypper install sysstat
Loading repository data...
Reading installed packages...
Resolving package dependencies...
```

The following NEW package is going to be installed:
sysstat

```
1 new package to install.
Overall download size: 174.0 KiB. After the operation, additional 662.0 KiB will
be used.
Continue? [y/n/?] (y): y
Retrieving package sysstat-8.1.5-7.45.24.s390x (1/1), 174.0 KiB (662.0 KiB
unpacked)
Installing: sysstat-8.1.5-7.45.24 [done]
# rpm -qa|grep sysstat
sysstat-8.1.5-7.45.24
```

iostat command

In addition to reporting overall system performance, the **iostat** command provides detailed I/O statistics for devices. Figure 9-10 on page 267 shows example output of the **iostat -x 5 2** command. It displays two samples with a 5-second interval.

Linux 3.10.0-229.el7.s390x (linux1.itso.ibm.com)						04/29/2015		_s390x_ (2 CPU)					
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	0.01	0.00	0.04	0.01	0.00	99.93							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
dasda	0.07	0.02	0.41	0.15	9.76	1.51	40.02	0.00	5.99	4.27	10.65	0.31	0.02
dm-0	0.00	0.00	0.02	0.00	0.06	0.00	8.00	0.00	1.10	1.10	0.00	0.50	0.00
dm-1	0.00	0.00	0.42	0.17	9.56	1.41	37.13	0.00	6.07	4.18	10.92	0.28	0.02
dasdb	0.04	0.00	0.02	0.00	0.08	0.00	9.25	0.00	0.06	0.06	0.00	0.06	0.00
dasdc	0.04	0.00	0.02	0.00	0.08	0.00	9.29	0.00	0.06	0.06	0.00	0.06	0.00
sdb	0.00	0.00	0.02	0.00	0.07	0.00	8.02	0.00	0.23	0.24	0.00	0.23	0.00
sda	0.00	0.00	0.02	0.00	0.08	0.00	8.26	0.00	0.32	0.30	2.50	0.29	0.00
sdc	0.00	0.00	0.02	0.00	0.07	0.00	8.18	0.00	0.20	0.20	0.00	0.20	0.00
sdd	0.00	0.00	0.02	0.00	0.07	0.00	7.98	0.00	0.23	0.23	0.00	0.20	0.00
dm-2	0.00	0.00	0.03	0.00	0.14	0.00	8.30	0.00	0.23	0.23	0.00	0.23	0.00
dm-3	0.00	0.00	0.00	0.00	0.02	0.00	8.00	0.00	0.49	0.49	0.00	0.49	0.00
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	0.00	0.00	0.00	0.00	0.00	100.00							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
dasda	0.00	0.00	0.00	0.20	0.00	0.30	3.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-1	0.00	0.00	0.00	0.20	0.00	0.30	3.00	0.00	0.00	0.00	0.00	0.00	0.00
dasdb	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dasdc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sdb	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sdc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sdd	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 9-10 iostat command output

On Figure 9-10, the avgqu-sz column shows the average queue size for a specific device. The larger the number, the more contention exists for the device.

The await column displays the average wait time for a device. It includes the time that requests spent in the queue and the time that was spent servicing requests in a storage device. The higher the number, the more time is wasted by a program that is waiting for I/O.

sar and sadc commands

Ad hoc data can be gathered by calling the data collector with the **sadc** command:

```
# /usr/lib64/sa/sadc -S ALL -F 5 60 /tmp/sadc.out
```

Data will be collected for 5 minutes in 5-second intervals (5 x 60 = 300 s = 5 minutes).

Output that is produced by the **sadc** command is a binary file. To process it and generate a text output, the **sar** command is used:

```
# sar -A -f /tmp/sadc.output > outfile.txt
```

The **sar** command generates detailed performance information. If **sar** is configured as a service and it gathers data automatically, its data is stored in the `/var/log/sa` directory. Data files are of the form `sa<dd>` where *dd* is the day of the month. Text files are of the form `sar<dd>`, for example:

```
# cd /var/log/sa
# file *
sa09: data
sa10: data
sa11: data
sa12: data
sa13: data
sar09: ASCII text
sar10: ASCII text
sar11: ASCII text
sar12: ASCII text
```



Working with disks

“Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.”

— Albert Einstein

This chapter relates to working with disks. Both extended count key data (ECKD) DASD and Fibre Channel Protocol (FCP)/Small Computer System Interface (SCSI) tasks are described. This chapter concentrates on performing tasks on Linux. For the z/VM perspective, see 6.2, “Considerations for disk storage types” on page 178.

10.1 Add disk space to virtual machines

This section describes how to add additional disk space to a Linux virtual machine. This disk space might come from different types of disks. The types of disk are described in 6.2, “Considerations for disk storage types” on page 178.

Important: If you add minidisks to the user directory for a certain virtual machine, it is possible to attach them to a running Linux system without “bouncing” it.

For example, if you added a minidisk at virtual address 104, you can use the following commands to link to the disk and then enable it:

```
# vmcp link '* 104 104 mr'
# chccwdev -e 104
```

10.1.1 Make new minidisks or count key data DASD available in Linux

After you get new minidisks or count key data (CKD) DASD, for example, at the addresses 0.0.0102, 0.0.0103, and 0.0.0104, make the new disks available by performing the following steps:

1. Make the disks visible with the **cio_ignore** command:

```
# cio_ignore -r 102
# cio_ignore -r 103
# cio_ignore -r 104
```

2. Depending on your operating system, follow these steps:

- If you use Red Hat Enterprise Linux Server 7.1 (RHEL), perform these tasks:

- i. Enable the disks with the **chccwdev -e** command:

```
# chccwdev -e 102 103 104
Setting device 0.0.0102 online
Done
Setting device 0.0.0103 online
Done
Setting device 0.0.0104 online
Done
```

- ii. Make a backup of **/etc/dasd.conf**, and then add minidisks 102, 103, and 104 to it:

```
# cd /etc
# cp dasd.conf dasd.conf.orig
# vi dasd.conf
0.0.0301
0.0.0300
0.0.0101
0.0.0100
0.0.0102
0.0.0103
0.0.0104
```

- If you use SUSE Linux Enterprise Server (SLES) 12, use the **dasd_configure** command to enable minidisks 102, 103, and 104:

```
# dasd_configure 0.0.0102 1
Configuring device 0.0.0102
Setting device online
# dasd_configure 0.0.0103 1
Configuring device 0.0.0103
Setting device online
# dasd_configure 0.0.0104 1
Configuring device 0.0.0104
Setting device online
```

3. View the available disks again with the **lsdasd** command:

```
# lsdasd
Bus-ID      Status      Name      Device  Type  BlkSz  Size      Blocks
=====
0.0.0301    active     dasda     94:0    FBA   512    512MB     1048576
0.0.0300    active     dasdb     94:4    FBA   512    256MB     524288
0.0.0100    active     dasdc     94:8    ECKD  4096   3521MB    901440
0.0.0101    active     dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0102    n/f        dasde     94:16   ECKD
0.0.0103    n/f        dasdf     94:20   ECKD
0.0.0104    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 10.2.1, “Create a logical volume and file system” on page 276. If you are extending an existing logical volume, skip ahead to 10.3, “Extend an existing logical volume” on page 280.

10.1.2 Make new emulated DASD (EDEV) available in Linux

For new emulated DASD (EDEV), for example, at address 0.0.0150, 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. Follow these steps:

1. Make the disk visible with the **cio_ignore** command:

```
# cio_ignore -r 150
```

2. Depending on your operating system, follow these steps:

- If you use Red Hat Enterprise Linux 7.1, follow these steps:

- i. Enable the disks with the **chccwdev -e** command:

```
# chccwdev -e 150
Setting device 0.0.0150 online
Done
```

- ii. Make a backup of `/etc/dasd.conf`, and then add the DASD 150 to it:

```
# cd /etc
# cp dasd.conf dasd.conf.orig
# echo 0.0.0150 >> dasd.conf
```

- If you use SUSE Enterprise Linux 12, use the **dasd_configure** command to enable the DASD 150:

```
# dasd_configure 0.0.0150 1
Configuring device 0.0.0150
Setting device online
```

3. View the available disks again with the **lsdasd** command:

```
# lsdasd
Bus-ID      Status      Name      Device  Type  BlkSz  Size      Blocks
=====
0.0.0301    active     dasda     94:0    FBA   512    512MB     1048576
0.0.0300    active     dasdb     94:4    FBA   512    256MB     524288
0.0.0100    active     dasdc     94:8    ECKD  4096   3521MB    901440
0.0.0101    active     dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0150    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 10.2.1, “Create a logical volume and file system” on page 276. If you are extending an existing logical volume, skip ahead to 10.3, “Extend an existing logical volume” on page 280.

10.1.3 Make 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 12 and RHEL 7.1 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 6.2.2, “Direct-attached Fibre Channel” on page 180. 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 7.1, 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:

```
# ls1uns
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, perform these tasks:

- 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 (36005076305ffc74c00000000000001017) 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 12, 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:

```
# ls1uns
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 already 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
```

10.2 Add 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.

10.2.1 Create 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 10-1 on page 277 shows a block diagram of the LVM.

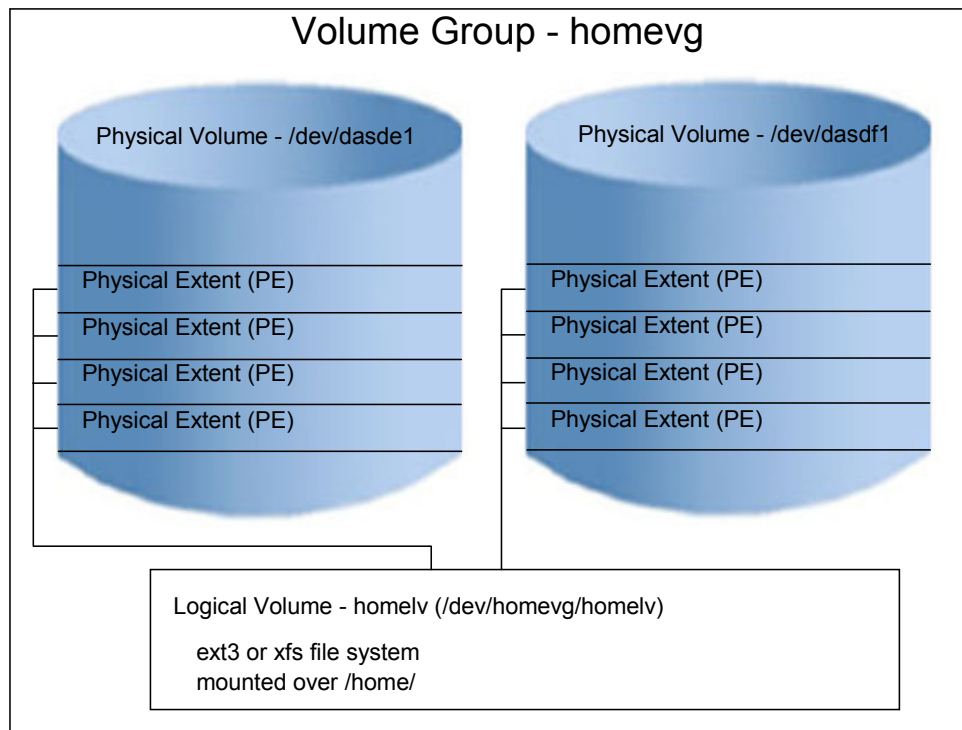


Figure 10-1 LVM block diagram

Create physical volumes from two minidisks

To create physical volumes from new minidisks at virtual device addresses 102 and 103, perform the following steps:

1. Check the devices on your system with the **lsdasd** command.
2. The **pvccreate** command initializes partitions for use by LVM. Initialize the two new DASD partitions:

```
# pvccreate /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           s0ugf1-h1V3-fYnf-1adW-4m0I-4HTJ-HdA0TU

"/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	v02PJY-gy4x-M9Hj-kt51-T04J-B4n5-Ntvkje

Create 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).

Create 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 home1v** 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 home1v homevg
Using default stripesize 64.00 KiB
Logical volume "home1v" 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/home1v
--- Logical volume ---
LV Path                /dev/homevg/home1v
LV Name                 home1v
VG Name                 homevg
LV UUID                 qNcyDp-Eeqs-gfB1-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
```

Make 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/home1v
...
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/home1v
...
```

The file system that was created from the logical volume is now ready to be mounted.

10.2.2 Update 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/home1v    /home                ext4    defaults    0 0
... // For SLES:
/dev/homevg/home1v    /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-home1v
                        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/`.

10.3 Extend an existing 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 10.1, “Add disk space to virtual machines” on page 270.

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-b1WP-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
=====
0.0.0100    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
0.0.0102    active      dasde     94:16   ECKD  4096   3521MB    901440
0.0.0103    active      dasdf     94:20   ECKD  4096   3521MB    901440
0.0.0104    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 **pvccreate** command:

```
# pvccreate /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-b1WP-3Kca-0xCI-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.

10.4 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 `home1v`:

```
# 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 `home1v` and all physical extents are used:

```
# lvcreate -l 1760 -n home1v homevg
Logical volume "home1v" created
```

4. Create a file system from the logical volume. In this example, it is type `ext4`:

```
# mkfs.ext4 /dev/homevg/home1v
```

5. Add the new file system to the file system table and mount it:

```
# vi /etc/fstab
...
# grep home /etc/fstab
/dev/homevg/home1v      /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-home1v 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-eXh0-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-eXh0-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-eXh0-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-pVZ03g
```

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.



Working with networks

“Two things are infinite: the universe and human stupidity; and I’m not sure about the universe.”

— Albert Einstein

This chapter describes the following miscellaneous tasks that you might need to perform:

- ▶ Adding channel-to-channel adapters (CTCAs) to a single system image (SSI) cluster
- ▶ Setting up a private interconnect
- ▶ Creating a HiperSockets device between logical partitions
- ▶ Configuring a port group with Link Aggregation Control Protocol

11.1 Adding CTCAs to an SSI cluster

The SSI CTC installation panel allows two CTC connections to be installed for each SSI member. You need to 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 will be available in a Fibre Channel connection (FICON) CTC control unit. It is recommended for performance reasons that only four of the eight devices are used.

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 (ITS0ZVM1):

```
===> q ctc active
CTCA 47E0 ATTACHED TO SYSTEM -ISFC
CTCA 57E0 ATTACHED TO SYSTEM -ISFC
```

- Display the installed CTCs on ITS0ZVM2:

```
===> 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 with the following commands:

- Display the channel paths that are used by the CTCs on ITS0ZVM1:

```
===> 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 ITS0ZVM2:

```
===> 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 performing the following steps:

1. Display the devices that are used by the channel paths on ITS0ZVM1:

```
===> 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 ITS0ZVM2:

```
===> 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 ITS0ZVM1:

```
===> 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 ITS0ZVM2:

```
===> 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.

11.1.1 Add the CTC devices dynamically

To add the CTC devices dynamically, perform the following steps:

1. Log on to MAINT on the first member.

2. Activate six CTCs on the first member, ITS0ZVM1:

```
===> 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 ITS0ZVM2:

```
===> 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 additional CTCs if you reissue the **QUERY CTC** command.

4. Issue the **QUERY CTC** command from ITS0ZVM1:

```
===> 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 ITS0ZVM2:

```

===> 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.

11.1.2 Add the CTC devices permanently

To add the CTC devices to the SSI permanently, perform 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 with 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 this example *before* the changes are made:

```

/*****
/*          Activate ISLINK statements          */
/*****

ITS0ZVM1:      ACTIVATE ISLINK 47E0 57E0  NODE ITS0ZVM2
ITS0ZVM2:      ACTIVATE ISLINK 4120 5120  NODE ITS0ZVM1

```

The SYSTEM CONFIG file looks like this example *after* the changes are made:

```

/*****
/*          Activate ISLINK statements          */
/*****

ITS0ZVM1:      BEGIN
                ACTIVATE ISLINK 47E0 47E1 47E2 47E3  NODE ITS0ZVM2
                ACTIVATE ISLINK 57E0 57E1 57E2 57E3  NODE ITS0ZVM2
ITS0ZVM1:      END
ITS0ZVM2:      BEGIN
                ACTIVATE ISLINK 4120 4121 4122 4123  NODE ITS0ZVM1
                ACTIVATE ISLINK 5120 5121 5122 5123  NODE ITS0ZVM1
ITS0ZVM2:      END

```


When the system is restarted, the ISLINKs are active between members.

11.2 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 (CEC) SSI cluster, the private interconnect must be able to connect the network on the guests. 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. Follow these 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 only to 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.

11.3 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:

- ▶ Verify HiperSockets hardware definitions
- ▶ Create a TCP/IP stack on z/OS
- ▶ Verify HiperSockets hardware definitions
- ▶ Verify connectivity

11.3.1 Verify 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 2-2 on page 31, 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.

11.3.2 Create 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 the *IBM HiperSockets Implementation Guide*, SG24-6816, at the following website:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg246816.pdf>

Follow these steps to create a TCP/IP stack on z/OS:

1. Create a TCP/IP stack (which is called TCPIP in this example) with a TCP/IP profile that uses the **F0** CHPID:

```
VIEW      TCPIP.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 IUTIQDF0
```

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 **TCPIP** address space that uses this profile.
3. Issue the command **D TCPIP,TCPIP,NETSTAT,DEVL** to verify the link information.

11.3.3 Configure the HiperSockets interface on Linux

We describe how 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 DEVTYP HIPER      CHPID F0 IQD
OSA 7004 ATTACHED TO LINUX2 E000 DEVTYP HIPER      CHPID F0 IQD
OSA 7005 ATTACHED TO LINUX2 E001 DEVTYP 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

Perform 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
encw0.0.e000
```

4. Create a network configuration file by using the `nmcli` command:

```
# nmcli con add type ethernet con-name hipersocket ifname encw0.0.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.e002"
# 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.

11.3.4 Verify 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 VIR13: Pinging host 10.1.1.43
Ping #1 response took 0.000 seconds.
```

2. Ping from the Red Hat Enterprise Linux that runs on ITS0ZVM1:

```
[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 ITS0ZVM2:

```
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.

11.4 Configuring a port group with Link Aggregation Control Protocol

To aggregate multiple OSA-Express ports, *port groups* can be defined on z/VM and attached to a virtual switch. Connectivity by using a port group 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 11-1.

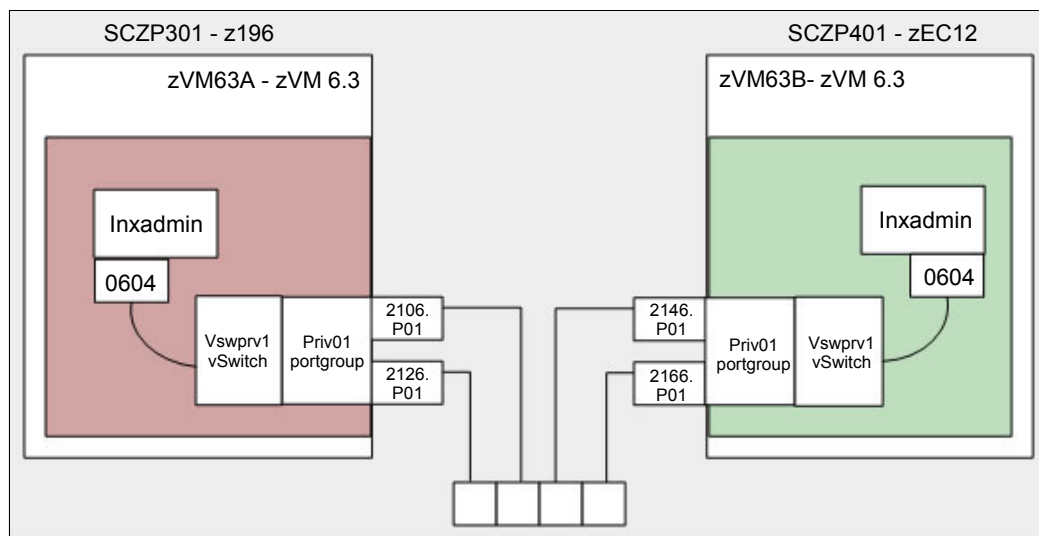


Figure 11-1 Port group Priv01 connectivity

Note: Port number 1, not port number 0, was used for this connection.

ITS0ZVM1 port group priv1 has the following details:

CHPID 00 portnumber 1 OSA device 2106

CHPID 01 portnumber 1 OSA device 2126

ITS0ZVM2 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 (ITS0ZVM1 in this example):

```
===> set port group priv01 join 2106.p01 2126.p01
```

2. Create the port group on the second SSI member (ITS0ZVM2 in this example):

```
===> set port group priv01 join 2046.p01 2066.p01
```

Note: Link Aggregation Control Protocol (LCAP) is set as active, by default. To use LACP, the network switch needs LACP to be active on the ports that the CHPIDs connect to.

Important: 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 input/output 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 (ITS0ZVM1 and ITS0ZVM2 in this example):

```
===> define vswitch vswprv1 rdev none ethernet vlan aware group priv01 gvrp
```

4. Create a virtual NIC on the Linux image to use the switch. If you are running DirMaint, the following command creates a network device and adds it to LNXADMIN:

```
===> dirm for lnxadmin nicdef 604 type qdio lan system vswprv1
```

5. If you are not running a directory maintenance product, you need to update the user directory entry for LNXADMIN with the following **DEFINE NIC** and **COUPLE** commands:

```
COMMAND SET VSWITCH VSWPRV1 GRANT &USERID
```

```
COMMAND DEFINE NIC 604 TYPE QDIO
```

```
COMMAND COUPLE 604 TO SYSTEM VSWPRV1
```

6. If RACF is running without DIRM, you need to permit the virtual switch VSWPRV1 access to LNXADMIN with the following commands:

- a. Define the system virtual switch that is named VSWPRV1 to the VMLAN class:

```
===> rac rdefine vmlan system.vswprv1
```

- b. Permit TCP/IP to the virtual switch VSW1:

```
===> rac permit system.vswprv1 class(vmlan) id(lnxadmin) access(update)
```

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.

11.5 Network commands

Regardless of which Linux distribution you are using, there are some commands which you should be aware of for interacting with network functionality.

Those who have worked with Linux (or Unix) for several years may be used to older commands, bundled with the `net-tools` package, most of which are actually deprecated.

The `iproute2` suite has 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. A brief listing of some commonly used commands is shown in Table 11-1 below. Many of the modern commands can be abbreviated; the full command is shown with parenthesis around the optional full-length version:

Table 11-1 Linux networking commands

Obsolete commands	Modern commands	Purpose
arp	ip n(eighbor)	Neighboring IP information
ifconfig	ip a(address) ip l(link) 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 l 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

More information on `iproute2` can be found at the following sites:

<http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2>

<http://www.policyrouting.org/iproute2-toc.html>

More information on the deprecated `net-tools` package can be found at the following site:

<http://www.linuxfoundation.org/collaborate/workgroups/networking/net-tools>



Miscellaneous helpful information

“Try not to become a man of success, but rather try to become a man of value.”

— Albert Einstein

This chapter contains miscellaneous information. These topics facilitate administration, save time, increase functionality, or add capabilities to your systems.

12.1 Install 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 both IBM employees, IBM Business Partners, and clients 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:

ibm.com/vm/download/license.html

12.1.1 Use the CMS web browser

To use the CMS web browser, which is called Charlotte, it must be re-blocked and then unpacked. In 4.6.3, “Copy the utilities and REXX EXECs to Shared File System (SFS) pools” on page 83, the VMARC was placed at VMPSFS:VMWW2. Follow these steps:

1. Log on as either MAINT or MAINT630 on the first node in the cluster.
2. Access VMPSFS:VMWW2 as W read/write and VMPSFS:MAINT630.UTILS.VMARC as V:

```
===> access vmpsfs:vmww2. W (forcerw  
===> access vmpsfs:maint630.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 information into a Shared File System (SFS) directory under MAINT or MAINT630 because *you must not access the web from a master system management virtual machine*, such as MAINT or MAINT630. 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
```

12.2 Manually formatting DASD for use

Often, you will need to add a DASD to the system for additional 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.

This example covers the formatting of three additional DASDs:

```
====> query 30D0-30D2
DASD 30D0 VM30D0 , DASD 30D1 VM30D1 , DASD 30D2 VM30D2
====> attach 30D0-30D2 to *
30D0-30D2 ATTACHED TO MAINT630
```

Allocate 30D0 to ITS0ZVM2 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



Part 3

Appendixes

This section consists of the following appendixes:

- ▶ Appendix A, “References, cheat sheets, and blank worksheets” on page 307
- ▶ Appendix B, “Additional material” on page 321



A

References, cheat sheets, and blank worksheets

This appendix refers to additional materials that are included for your reference. These materials can either be printed or downloaded from the Internet as described.

Related books

The following publications can be used as information sources:

- ▶ *Documentation for z Systems Linux Development stream*, which is available on the web at the following websites:
 - http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html
 - http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html
- ▶ z/VM Internet Library online documentation:
<http://www.vm.ibm.com/library/>

Under the link that is labeled z/VM V6.3 PDF Files, the following books are useful:

- *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

Under the link that is labeled Program Directories, the following books are useful:

- *Performance Toolkit for VM*
- *DirMaint*
- *RACF Security Server for z/VM*
- ▶ SUSE Linux Enterprise Server 12 on IBM z Systems:
https://www.suse.com/documentation/sles-12/book_sle_deployment/data/cha_zseries.html
- ▶ IBM Redbooks publications at <http://www.redbooks.ibm.com/>:
 - *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

These websites and URLs are also relevant as further information sources:

- ▶ 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/vmsslinfo.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>
- ▶ 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 A-1 on page 310 provides a summary and the location of important z/VM configuration files.

Table A-1 Important z/VM configuration files

File	Location	Description
SYSTEM CONFIG	PMAINT CFO	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 TCP/IP	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 TCP/IP.
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.
TCP/IP 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 XEDIT 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.

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:

1. 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.
2. 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).
hkl;	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.

3. 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: 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.

Tip: 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 2.1, “Planning for VMSSI with LGR” on page 16. 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.

IBM Shopz

If you are ordering z/VM by using *Shopz*, as described in 4.1, “Obtaining z/VM through electronic download” on page 52, use Table A-2 to record the values that you will use.

Table A-2 *Shopz* data

Name	Value	Comment
Starting URL	ibm.com/shopz	

Name	Value	Comment
User ID		Customer number (For IBM employees, it is your intranet user ID and password.)
Password		
Order number		

Hardware Management Console

In 4.3.1, “Start the z/VM installation” on page 56, we describe how to start a z/VM installation from the Hardware Management Console (HMC). Complete Table A-3 to record the values that you will use.

Table A-3 HMC values

Name	Value
HMC host name or URL	
HMC user ID	
HMC password	
FTP source system (if you are installing from FTP)	
z/VM installation directory	

z/VM Installation Planning Panels (INSTPLAN)

In 4.3.2, “Copy a plain z/VM system to DASD” on page 59, we describe the INSTPLAN command that is run from the Integrated 3270 Console. The following information will be necessary.

INSTPLAN panels 1 and 2

Complete Table A-4 to record the values that are required in the first two INSTPLAN panels.

Table A-4 INSTPLAN values for the first two panels

Name	Value	Comment
Language	<input type="checkbox"/> AMENG <input type="checkbox"/> USENG <input type="checkbox"/> KANJI	AMENG (American English), UCENG (uppercase English), or KANJI.
DASD model	<input type="checkbox"/> 3390 Model-3 <input type="checkbox"/> 3390 Model-9	3390 Model-3 or Model-9 (Installation to a fixed-block architecture (FBA) disk is not described in this book.).
File pool name		VMPSYS (default) recommended.
System type		Single system image (SSI) (Non-SSI is not described in this book.).
Non-SSI system name		Used for non-SSI installation only.
Number of members		SSI installation only (usually 2 or 4).

Name	Value	Comment
SSI cluster name		SSI installation only.
Automatic configuration		<i>“No” is strongly recommended.</i>

INSTPLAN panel 3

Complete Table A-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 A-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 A-6 to record the volume labels and real device addresses that are required in the Installation Volume Definition INSTPLAN panel.

Table A-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)

Type	Default label	Chosen label	Address	Comment
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
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 A-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 4.3.2, “Copy a plain z/VM system to DASD” on page 59.

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 A-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 A-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 A-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 <i>triplet</i> 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 A-9 to document the z/VM DASD that you will use.

Table A-9 z/VM DASD blank worksheet

Device number	Label	Type	Notes

Device number	Label	Type	Notes

Linux resources worksheet

Use the worksheet in Table A-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 A-10 Linux NFS server resources blank worksheet

Name	Value	Comment
TCP/IP address		
User/password		
NFS-exported installation directory		

Use the worksheet in Table A-11 to document your Linux on z Systems resources.

Table A-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 A-12 to document the host names and associated IP addresses and virtual machines that you will use.

Table A-12 Host names blank worksheet

Host name	IP address	Virtual machine/ LPAR	Notes



B

Additional material

This book refers to additional material that can be downloaded from the Internet.

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:

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:

<i>File name</i>	<i>Description</i>
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. The following sections are included:

- ▶ “z/VM REXX EXECs and XEDIT macros” on page 323
- ▶ “Sample files” on page 331
- ▶ “Linux code” on page 332

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 4.12, “Enabling basic system automation” on page 109.

```
/* **** */
/*
/* 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 "          <-----<                                "
    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)
            End
        End
        formatted = formatted label
    End
End

```

```

        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 reside 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 reside 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 OOC CLEAR'
Exit
```


SLES11S3 EXEC

This section lists the code for the `sles11s3 EXEC` that starts a SLES 11 SP3 installation. It is recommended to reside 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 OOC CLEAR'
Exit
*****/
```

SWAPGEN EXEC

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"
    fi
}

```

```

        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
}

```

```

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/lsmmod | 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
}

```

```

    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 command 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 s1lgold 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 '[:lower:]' '[:upper:]'` # 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 $3 }'`
    cmsfs1st -d /dev/$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 listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- ▶ *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux Server 7.1*, SG24-8303
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- ▶
- ▶ *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266
- ▶ *Security on z/VM*, SG24-7471
- ▶ *Running Linux Guests with less than CP Class G Privilege*, REP-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

These 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:
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

These websites are also relevant as further information sources:

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:
ibm.com/vm/techinfo/listserv.html

z/VM

- ▶ Installation
ibm.com/vm/install
- ▶ Publications:
ibm.com/vm/pubs
- ▶ Technical library:
ibm.com/vm/library
- ▶ Security
ibm.com/vm/security
- ▶ Performance:
ibm.com/vm/perf

- Performance tips:
ibm.com/vm/perf/tips

IBM Techdocs

- IBM Techdocs technical sales library:
ibm.com/support/techdocs

Linux distributions

Red Hat

- Documentation for z Systems Linux Development stream:
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:
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:
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 Global 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 roadmaps and improve communication between end-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 at the following URL:

ibm.com/developerworks/rfe

Redbooks

The Virtualization Cookbook for IBM z Systems Volume 1:

SG24-8147-01

ISBN 0738440868



(0.5" spine)

0.475" <-> 0.873"

250 <-> 459 pages



SG24-8147-01

ISBN 0738440868

Printed in U.S.A.

Get connected

