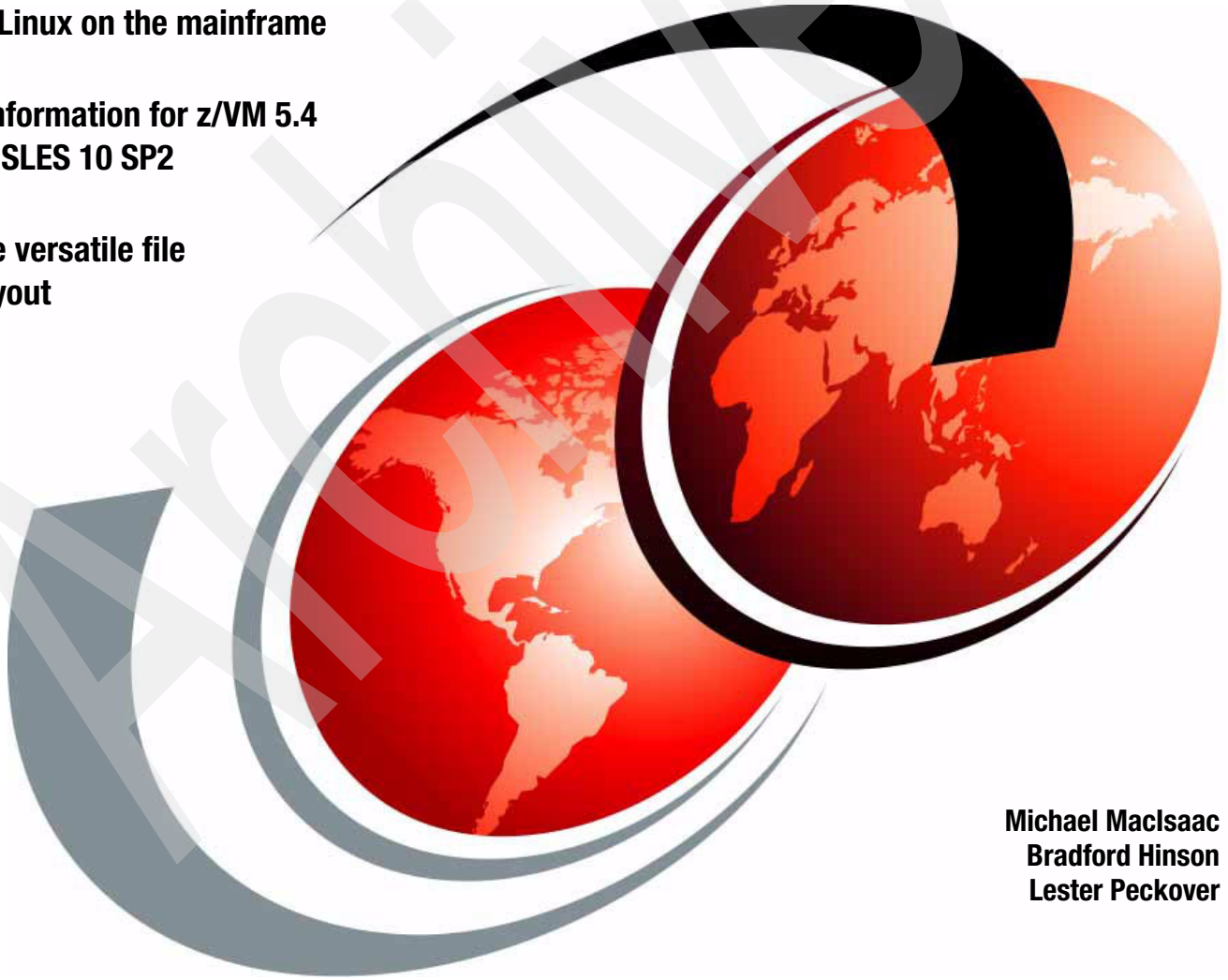
IBM

z/VM and Linux on IBM System z
The Virtualization Cookbook for SLES 10 SP2

- Hands-on instructions for installing z/VM and Linux on the mainframe
- Updated information for z/VM 5.4 and Linux SLES 10 SP2
- New, more versatile file system layout



Michael MacIsaac
Bradford Hinson
Lester Peckover

Redbooks

ibm.com/redbooks

Archived

Archived

International Technical Support Organization

The Virtualization Cookbook for SLES 10 SP2

October 2008

Archived

SG24-7493-00

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

Archived

First Edition (October 2008)

This edition applies to Version 5, Release 4, Modification 0 of z/VM (product number 5741-A05) and Novell SUSE Linux Enterprise Server (SLES) Version 10, Service Pack 2 (SP2).

© Copyright International Business Machines Corporation 2008. 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

Preface xi
The team that wrote this book xi
History xii
Summary of changes in October 2008 version xiii
Chapters and appendixes xiv
Conventions xv
Become a published author xv
Comments welcome xv

Chapter 1. Introduction to z/VM and Linux 1
1.1 What is virtualization 2
1.2 This book's approach 3
1.3 Choices and decisions made in this book 3
1.4 Infrastructure design 4
1.5 Usability tests performed for this book 5

Chapter 2. Planning 7
2.1 Bill of materials 8
2.1.1 Hardware resources 8
2.1.2 Software resources 8
2.1.3 Networking resources 9
2.2 z/VM conventions 9
2.2.1 Volume labeling convention 9
2.2.2 Backup file naming convention 10
2.2.3 The command retrieve convention 10
2.3 Disk planning 10
2.4 Memory planning 12
2.5 Password planning 12
2.6 Planning worksheets 13
2.6.1 z/VM resources used in this book 13
2.6.2 z/VM DASD used in this book 14
2.6.3 Linux resources used in this book 15
2.6.4 Linux user IDs used in this book 15
2.7 Blank worksheets 16
2.7.1 z/VM resources worksheet 16
2.7.2 z/VM DASD worksheet 17
2.7.3 Linux resources worksheet 18
2.7.4 Linux user ID worksheet 18

Chapter 3. Configuring a desktop machine 19
3.1 PuTTY: a free SSH client for Windows 20
3.2 Setting up a VNC client 24
3.2.1 Downloading and running RealVNC 24
3.2.2 Customizing RealVNC 25

© Copyright IBM Corp. 2008. All rights reserved.

3.3 3270 emulators 25

Chapter 4. Installing and configuring z/VM 27
4.1 Installing z/VM from DVD 28
4.1.1 Booting z/VM from DVD 28
4.1.2 Copying a vanilla z/VM system to DASD 31
4.1.3 IPL the vanilla z/VM from DASD 34
4.1.4 Completing the z/VM installation 36
4.2 Configuring TCP/IP 38
4.2.1 Use the IPWIZARD tool 38
4.3 Configuring the XEDIT profile 41
4.4 Customizing the SYSTEM CONFIG file 41
4.5 Configuring TCP/IP to start at IPL time 44
4.5.1 Renaming the TCPIP configuration file 45
4.5.2 Copy the PROFILE XEDIT file 45
4.5.3 Configuring the FTP server 46
4.5.4 Shutting down and reIPLing the system 47
4.6 Adding paging volumes 47
4.6.1 Formatting the paging volumes 48
4.6.2 Formatting DASD for minidisks 50
4.6.3 Updating the SYSTEM CONFIG file 52
4.6.4 Testing the changes 53
4.7 Creating a user ID for common files 54
4.7.1 Define the user in the USER DIRECT file 54
4.7.2 Logging and customizing the new user ID 56
4.7.3 Copying a PROFILE XEDIT 57
4.7.4 Creating a PROFILE EXEC 57
4.7.5 Copying files associated with this book to LNXMAINT 58
4.8 Customizing system startup and shutdown 59
4.8.1 Configuring the AUTOLOG1 PROFILE EXEC 59
4.8.2 Testing the changes 60
4.9 Addressing z/VM security issues 60
4.9.1 VM security products 61
4.9.2 High level z/VM security 61
4.9.3 Changing passwords in USER DIRECT 61
4.10 Backing up your z/VM system to tape 62
4.11 Relabeling the system volumes 63
4.11.1 Modifying labels in the SYSTEM CONFIG file 64
4.11.2 Modifying labels in the USER DIRECT file 66
4.11.3 Changing the labels on the five volumes 66
4.11.4 Shutting down your system and restarting it 67
4.12 Restoring your z/VM system from tape 68

Chapter 5. Servicing z/VM 71
5.1 Applying a PTF 72
5.1.1 Determining whether a PTF has been applied 72
5.1.2 Getting service through Internet FTP 73
5.1.3 Downloading the service to z/VM 74
5.1.4 Receiving, applying, and building service 75
5.1.5 Putting the service into production 76
5.2 Applying a Recommended Service Upgrade 77
5.2.1 Getting service from the Internet 77
5.2.2 Downloading the service to z/VM 78

5.2.3 Receiving, applying, and building the service 80
5.2.4 Putting the service into production 81
5.3 Determining the z/VM service level 82

Chapter 6. Configuring an NFS server for SLES 10 SP2 85
6.1 Getting started 86
6.2 Downloading files associated with this book 86
6.3 Setting up a SLES 10 SP2 install server 86
6.3.1 SLES 10 SP2 DVD ISO image file 86
6.3.2 Starting from physical DVD 87
6.3.3 Verifying the ISO images 87
6.3.4 Configuring the SLES 10 SP2 install server 88
6.4 Enabling the NFS server 88

Chapter 7. Installing SLES10 SP2 on the controller 91
7.1 Creating the user ID LINUX00 92
7.2 Updating AUTOLOG1 PROFILE EXEC 94
7.3 Preparing SLES10 SP2 bootstrap files 94
7.4 Installing the controller 97
7.4.1 Verifying the installation 107
7.5 Configuring the controller 108
7.5.1 Copying files to the controller 108
7.5.2 Resetting the install source location 109
7.5.3 Configuring the NFS server 110
7.5.4 Turning off unneeded services 110
7.5.5 Applying service if necessary - online update 111
7.5.6 Installing the cmsfs package 111
7.5.7 Enabling the vmcp and cmm modules 112
7.5.8 Setting system to halt on SIGNAL SHUTDOWN 113
7.5.9 Modifying zipl.conf 113
7.5.10 Rebooting the system 114
7.5.11 Verifying the changes 115

Chapter 8. Installing SLES10 SP2 on the golden image 117
8.1 Creating the SLES10S2 user ID 118
8.2 Creating the SLES10S2 parameter file 119
8.3 Updating AUTOLOG1 PROFILE EXEC 119
8.4 Installing the golden image 120
8.4.1 Begin the SLES10 SP2 installation 121
8.4.2 Beginning YaST installation 122
8.4.3 Booting your new Linux system from disk 129
8.4.4 Completing YaST2 installation 130
8.5 Configuring the golden image 132
8.5.1 Configuring the VNC server 132
8.5.2 Preparing for Online Update 135
8.5.3 Turning off unneeded services 135
8.5.4 Enabling the cmm module 136
8.5.5 Applying service - Online Update 136
8.5.6 Configuring /etc/inittab 138
8.5.7 Verifying the On Demand Timer Patch is set 138
8.5.8 Configuring SSH keys 139
8.5.9 Modifying zipl.conf 140
8.5.10 Cleaning temporary files 141
8.5.11 Rebooting the system and verifying changes 141

Archived

Chapter 9. Cloning SLES 10 SP2 143
9.1 Formatting DASD for minidisks 144
9.2 Defining a new user ID for a virtual server 145
9.2.1 Adding LINUX01 to AUTOLOG1 PROFILE EXEC 146
9.2.2 Creating a parameter file for the new LINUX ID 147
9.3 Cloning a virtual server manually 148
9.4 Cloning a virtual server automatically 153
9.5 Creating three more virtual servers 155
9.5.1 Defining three more user IDs 155
9.5.2 Creating three new parameter files 156
9.5.3 Granting user IDs access to VSWITCH 157
9.5.4 Testing logging on to a new user ID 158
9.6 Reviewing system status 158

Chapter 10. Cloning open source virtual servers 161
10.1 Creating a virtual Web server 162
10.1.1 Installing Apache RPMs 162
10.1.2 Testing Apache 162
10.1.3 Populating your Web site 162
10.1.4 Apache resources 163
10.2 Creating a virtual LDAP server 163
10.2.1 Cloning a Linux 163
10.2.2 Configuring the LDAP server 163
10.2.3 Adding an LDAP user 165
10.2.4 Setting another virtual server to use the LDAP server 167
10.3 Creating a virtual file and print server 169
10.3.1 Cloning a Linux virtual server 170
10.3.2 Installing the necessary RPMs 170
10.3.3 Configuring the Samba configuration file 170
10.3.4 Adding a Samba user 171
10.3.5 Starting Samba at boot time 171
10.3.6 Testing your changes 172
10.3.7 Configuring printing 173
10.4 Creating a virtual application development server 173
10.4.1 Cloning a Linux virtual server 174
10.4.2 Scripting languages 174
10.4.3 C/C++ development tools 175
10.4.4 Java development tools 176
10.4.5 Additional resources 177

Chapter 11. Miscellaneous recipes 179
11.1 Adding a logical volume 180
11.1.1 Adding minidisks to the z/VM directory entry 180
11.1.2 Making the new minidisks available 180
11.1.3 Formatting and partitioning the minidisks 181
11.1.4 Creating a logical volume and file system 181
11.1.5 Updating the file system table 184
11.2 Extending an existing logical volume 186
11.3 Centralizing home directories for LDAP users 188
11.3.1 Configuring the NFS server 189
11.3.2 Configuring the golden image for LDAP and automount 189
11.4 Setting the software clock through NTP 192
11.5 Rescuing a Linux system 194
11.5.1 Entering single user mode 194

11.5.2 Entering a rescue environment . . . 195
11.5.3 Integrated ASCII console . . . 197
11.6 Configuring rsync . . . 197
11.7 Enabling VMRM to work with CMM . . . 198
11.7.1 Configuring CMM1 . . . 199
11.7.2 How to observe that CMM1 is working . . . 200

Chapter 12. Monitoring z/VM and Linux . . . 201
12.1 Using INDICATE and other commands . . . 202
12.1.1 Using the INDICATE command . . . 202
12.1.2 Using other basic commands . . . 204
12.2 z/VM Performance Toolkit . . . 206
12.2.1 Configuring the z/VM Performance Toolkit . . . 206
12.2.2 Configuring Web Browser support . . . 207
12.2.3 Configuring PERFSVM . . . 208
12.2.4 Starting the z/VM Performance Toolkit . . . 210
12.2.5 Using the z/VM Performance Toolkit . . . 210
12.3 Monitoring Linux . . . 213
12.3.1 Monitoring Linux with rmfpms . . . 214
12.3.2 Monitoring Linux performance data from the kernel . . . 215
12.4 Viewing Linux data in the Performance Toolkit . . . 216
12.5 Common Performance Toolkit issue . . . 217

Appendix A. References . . . 219
z/VM configuration files . . . 220
Quick reference sheets . . . 220
XEDIT quick reference sheet . . . 220
vi quick reference sheet . . . 221

Appendix B. z/VM source code . . . 223
Obtaining and using the Web material . . . 223
z/VM REXX EXECs and XEDIT macros . . . 224
The CPFORMAT EXEC . . . 224
The CHPW540 XEDIT macro . . . 228
The LABEL540 EXEC . . . 231
The LABEL540 XEDIT macro . . . 233
PROFILE EXEC for Linux user IDs . . . 234

Appendix C. Linux source code . . . 235
The clone.sh script . . . 236

Related publications . . . 243
IBM Redbooks publications . . . 243
Other publications . . . 243
Online resources . . . 244
How to get Redbooks . . . 244
Help from IBM . . . 244

Index . . . 245

Archived

Archived

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 give 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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.

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.

© Copyright IBM Corp. 2008. All rights reserved.

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 http://www.ibm.com/legal/copytrade.shtml

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

AIX®
DirMaint™
ECKD™
IBM®
OMEGAMON®
RACF®
Redbooks®
Redbooks (logo) ®
REXX™
RMF™
S/390®
System p®
System x™
System z10™
System z9®
System z®
Tivoli®
Workplace™
z/OS®
z/VM®
z10™
z9®
zSeries®

The following terms are trademarks of other companies:

Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Java, JRE, MySQL, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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

Archived

Preface

This IBM® Redbooks® publication describes how to “roll your own” Linux® virtual servers on IBM System z® hardware under z/VM®. This edition applies to Version 5, Release 4, Modification 0 of z/VM (product number 5741-A05) and Novell® SUSE® Linux Enterprise Server (SLES) Version 10, Service Pack 2 (SP2).

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 deployed in a matter of minutes. This powerful “build and clone” capability can enable you to launch new products and services without the exhaustive planning, purchasing, installing and configuring of new hardware and software that can be associated with conventional discrete hardware servers. Development groups that need test environments built and rebuilt rapidly to enable them to efficiently deliver their projects, while handling change management in the process, can also benefit from this unique advantage.

The publication adopts a cookbook format that provides you with a concise, repeatable set of procedures for installing and configuring z/VM in a logical partition (LPAR), and then installing and customizing Linux. To implement these instructions, you need an IBM System z logical partition with associated resources, z/VM 5.4 media, and a Linux distribution.

This book assumes that you have a general familiarity with System z technology and terminology. It does not assume an in-depth understanding of z/VM and Linux. It is written for those who want to get a quick start with z/VM and Linux on the mainframe.

The team that wrote this book

This book was originally written in 2005 by Michael MacIsaac, Jin Xiong and Curtis Gearhart. It was updated in 2006 by Michael MacIsaac, Carlos Ordonez and Jin Xiong. It was updated a third time in late 2006 and early 2007 by Marian Gasparovic taking the lead on the SLES 10 version and Brad Hinson of Red Hat working on the RHEL 5 version.

In late 2007 and early 2008, Michael MacIsaac, Brad Hinson, and Lester Peckover updated it for z/VM 5.4, SLES 10 SP2, and RHEL 5.2.

Michael MacIsaac supports Linux and z/VM on IBM System z® in Poughkeepsie, NY.

Bradford Hinson is the Sr. Support Engineering lead for System z at Red Hat in Raleigh, North Carolina. He specializes in Linux support for the mainframe, and leads technical oversight of all System z projects and activities. He holds a BS degree in Computer Science from the University of North Carolina at Chapel Hill, and MS® degrees from UNC Greensboro and UNC Chapel Hill.

Lester Peckover is an Infrastructure, Enterprise Performance, and Capacity Architect in the United Kingdom. He has 32 years of experience in IT covering IBM System x™ and System p®, and has also worked on a number of other system platforms in performance and other areas, specializing in virtualization. His areas of expertise include System z and Linux running on the mainframe, as well as performance analysis and modelling. He helped to develop the IBM VM Performance Toolkit and other IBM VM program products and tools from other vendors. He has written, taught, and presented worldwide on performance and capacity and other system management topics.

© Copyright IBM Corp. 2008. All rights reserved.

Production of this IBM Redbooks publication was managed by:

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 e-business development and database management technologies. Lydia is a Certified IT Specialist with an MBA in Technology Management and has been employed by IBM for more than 23 years in various technology areas.

Sincere thanks to the following people who contributed to this project in many different ways:

Jerry Epker
Fidelity Information Services

Bill Bitner, Bruce Hayden, Denny Refsnider, Roger Lunsford, Jim Switzer and Romney White
IBM Endicott

Carsten Otte, Claudia Prawirakusumah
IBM Germany

Ray Mansell
IBM Hawthorne

Kim Betros, Duane Beyer, Michel Considine, Roy Costa, Greg Geiselhart, Dionne Graff, Susan Greenlee, Ernest Horn, Rong Lee, Bill Norton, Lydia Parziale, Thang Pham, Bruce Smilowitz, Kyle Smith, Paul Sutera, Helen Tsang, Donna Von Dehsen, Hossee Wakil, Jeng-Fang Wang and Dennis Wunder
IBM Poughkeepsie

Bernard Wu
NCCI

Mark Post
Novell

Matthew Eernisse
OpenSource Applications

David Boyes, Adam Thornton
Sine Nomine Associates

Chester Hood
State of Tennessee

Thanks the zBMC team in IBM Poughkeepsie and to the many who answered questions on the linux-390 list server. Special thanks to Tony Giaccone and Roy Costa from IBM Poughkeepsie for always finding another free LPAR and more DASD.

History

There are a number of *Virtualization Cookbooks*, so a brief history follows.

Archived

2008

In October 2008, two cookbooks were published, one targeting the Novell/SUSE SLES distribution, the other targeting the Red Hat RHEL distribution:

- This book: *z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2*, SG24-7493
- *z/VM and Linux on IBM System z The Virtualization Cookbook for Red Hat Enterprise Linux 5.2*, SG26-7492

Associated with these Redbooks are REXX™ EXECs and Linux scripts to help you install and and configure z/VM and Linux. These tools are not IBM products and are not formally supported. However, they are informally supported and are available on the Web.

2007

In March 2007, two cookbooks were published on linuxvm.org/present, each book targeting a different distribution:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10*
- *z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5*

2006

In September 2006, a cookbook was published that addressed both 31-bit and 64-bit RHEL 4:

- *IBM z/VM and Linux on IBM System z: Virtualization Cookbook for Red Hat Enterprise Linux 4*, SG24-7272

In August 2006, a cookbook was published on linuxvm.org/present that addressed both 31-bit and 64-bit SLES 9:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook 2*

Summary of changes in October 2008 version

There are significant changes in this book:

- The z/VM sections are updated for V5.4.
- The Novell/SUSE Linux sections are updated for SLES 10 SP2.
- The Linux system that is cloned is called *golden image* instead of *master image*.
- The controller and the golden image are installed onto two separate user IDs, not on the same user ID as in previous books.
- The file system layout recommends two 3390-3s (100 and 101) instead of just one (100) and implements logical volumes for more adaptable file systems.
- Sections were added on Cooperative Memory Management (CMM1).

The release of this book also removes some sections:

- The DAZL application has been removed.
- The “Backup and Restore” chapter has been removed.

Archived

Chapters and appendixes

The contents of this book are summarized in the following list:

- Chapter 1, “Introduction to z/VM and Linux” on page 1, explains the concept of virtualization and z/VM infrastructure design, and describes the usability tests performed during the development of this book.
- Chapter 2, “Planning” on page 7, describes how to plan hardware, software, and networking resources. It discusses DASD labeling conventions used in the book and password planning. Sample worksheets are provided for the examples used in the book, along with blank copies for your use.
- Chapter 3, “Configuring a desktop machine” on page 19, describes how to set up Windows® desktops. Specifically, the following tools are discussed:
  - How to get and set up PuTTY: a commonly used SSH client
  - How to get and set up a VNC client: a tool for running graphical applications
  - 3270 emulator applications
- Chapter 4, “Installing and configuring z/VM” on page 27, shows how to install and configure z/VM.
- Chapter 5, “Servicing z/VM” on page 71, describes how to apply service to z/VM both in the form of Programming Temporary Fixes (PTFs) and Recommended Service Upgrades (RSUs).
- Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85, explains how to set up a temporary NFS server on a Linux PC for the purpose of installing the first two Linux images. After the System z controller Linux is installed, you can copy the Linux install tree to it and retire the Linux PC server.
- Chapter 7, “Installing SLES10 SP2 on the controller” on page 91, describes how to install and configure two Linux images onto the first Linux user ID: the *golden image*, which it is cloned from, and the *controller*, which does the cloning among other tasks.
- Chapter 8, “Installing SLES10 SP2 on the golden image” on page 117, illustrates how to move the Linux install tree from the Linux PC server to the controller under z/VM.
- Chapter 9, “Cloning SLES 10 SP2” on page 143, explains how to prepare z/VM user IDs and clone your first virtual server.
- Chapter 10, “Cloning open source virtual servers” on page 161, shows how to configure cloned Linux images into the following virtual servers:
  - Web server virtual server
  - LDAP virtual server
  - File and print virtual server
  - Basic application development system
- Chapter 11, “Miscellaneous recipes” on page 179, describes how to add a logical volume to a Linux system and how to set up a z/VM Discontiguous Saved Segment (DCSS) in conjunction with the Linux eXecute In Place 2 (xip2) file system.
- Chapter 12, “Monitoring z/VM and Linux” on page 201, describes basic steps to begin monitoring z/VM and your new Linux virtual servers.
- Appendix A, “References” on page 219, provides reference Web sites, books and other pertinent information.
- Appendix B, “z/VM source code” on page 223 lists the z/VM source code associated with this book: z/VM REXX EXECs and XEDIT macros.
- Appendix C, “Linux source code” on page 235 lists the one Linux script associated with this book.

Archived

Conventions

The following font conventions are used in this book:

Monospace and bold	These fonts indicate commands entered by the user on the command line
<value>	Values inside angle brackets are examples and are to be replaced with values correct for your enterprise.
monospace	This font indicates file, directory, user ID and minidisk names.

The following command conventions are used in this book:

- z/VM commands are prefixed with **==>**
- z/VM XEDIT subcommands are prefixed with **====>**
- Linux commands running as root are prefixed with #
- Linux commands running as non-root are usually prefixed with $

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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 in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Send your comments directly to:

  mikemac@us.ibm.com

Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099

Archived

2455 South Road
Poughkeepsie, NY 12601-5400

Archived

1

Introduction to z/VM and Linux

Virtualization is a hot topic in the IT industry. The IBM mainframe, z/VM and its predecessors have been performing virtualization for four decades. Today, it is the most functionally rich virtualization platform available. When Linux came to the IBM mainframe in 2000, it was a natural fit to run under z/VM. You can run many tens of Linux images on the same System z logical partition (LPAR). Some customers are running hundreds in production mode.

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 deployed in a matter of minutes. This powerful build and clone capability can enable you to launch new products and services without the exhaustive planning, purchasing, installing and configuring of new hardware and software that can be associated with conventional discrete hardware servers. Development groups who need test environments built and rebuilt rapidly to enable them to efficiently deliver their projects, while handling change management in the process, can also benefit from this unique advantage.

Listed here are some of the most significant strengths offered by the IBM mainframe and z/VM:

- Their virtualization capabilities are more mature and robust than any other hardware and hypervisor combination.
- z/VM provides a rich, functional, and sophisticated level of system management which can greatly benefit running large numbers of Linux servers.
- z/VM's virtual switch (VSWITCH) makes networking Linux much simpler.
- Full volume backup of systems allows for complete disaster recovery when another data center is available.
- z/VM is one of the easiest operating systems to customize at the base installation level, with only a relatively small number of configuration files. After it is set up, z/VM will run for months with little maintenance and administration required.

Much function has been added to z/VM since version 5.2, as described in the following section.

© Copyright IBM Corp. 2008. All rights reserved.

z/VM 5.3

z/VM 5.3 became generally available in June 2007. Scalability was extended to allow 256 GB of real memory, a total of 8 TB of virtual storage, and 32 real processors. z/VM V5.3 also added support for the Collaborative Memory Management Assist (CMMA) on the z9® EC and the z9 BC processors or later. Virtual Machine Resource Manager (VMRM) detects when memory is constrained and notifies the Linux guests, which can then adjust their memory consumption to help relieve the memory constraint.

In the previous major release (z/VM 5.2), many memory contention issues were removed with the Control Program (CP) now using memory above 2 GB for a much broader set of operations. Previously, guest pages had to be moved below 2 GB for many reasons, for example in both standard I/O and Queued Direct I/O (QDIO). Now I/O can be performed using buffers anywhere in real memory, and QDIO structures can reside above 2 GB, as can most CP control blocks. These improvements offer constraint relief for large-real-memory virtual server environments that are memory-intensive.

z/VM 5.4

z/VM 5.4, available in August 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 exploits new capabilities of the System z10™ including:

- Greater flexibility, with support for the new z/VM-mode logical partitions, allowing all System z processor-types (CPs, IFLs, zIIPs, zAAPs, and ICFs) to be defined in the same z/VM LPAR for use by various guest operating systems
- The capability to install Linux on System z from the HMC, which eliminates network setup or a connection between an LPAR and the HMC
- Enhanced physical connectivity by exploiting all OSA-Express3 ports, thus helping to service the network and reduce 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, thereby avoiding the need to shut down z/VM and its guests, deactivate the LPAR, change its memory allocation, reactivate the LPAR, re-IPL z/VM, and restart its guests. Memory can be added non-disruptively to individual guests that support the dynamic memory reconfiguration architecture.

Read more about System z virtualization capabilities on the Web at:

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

1.1 What is virtualization

Virtualization is the ability for a computer system to share resources so that one physical server can act as many *virtual* servers. z/VM allows the sharing of the mainframe's physical resources such as disk (DASD), memory (sometimes called storage), network adapters (OSA cards) and CPU (CPs or IFLs). These resources are managed by a *hypervisor*. The z/VM hypervisor is called Control Program (CP).

When a user logs onto z/VM, the hypervisor creates a virtual machine which can run one of many different operating systems. The two operating systems discussed in this book are the z/VM native operating system, known as the Conversational Monitoring System (CMS), and Linux. CMS can be thought of as a z/VM shell. Virtual machines running Linux as guests of a z/VM host become the virtual servers.

1.2 This book's approach

Today there are numerous technical publications that discuss virtualization, but few of them demonstrate how to achieve it. This book gives you the "hands on" instructions needed to achieve a virtualized environment on your enterprise.

The approach adopted is to keep all solutions simple, as expressed in the well-known quote from Albert Einstein "Everything should be made as simple as possible, but not simpler". This book attempts to use a clear and insightful presentation to explain the somewhat complex topics covered here.

1.3 Choices and decisions made in this book

When deciding on installing, maintaining and provisioning (cloning) Linux virtual servers under z/VM, there are many basic choices to make. Here are some of the choices and assumptions made in this book:

- Use of "roll your own" cloning versus cloning products

  A discussion of cloning products such as Aduva Onstage, the Mainstar Provisioning Expert, IBM Tivoli® Provisioning Manager and IBM Director is beyond the scope of this book. Although these are all viable solutions, the cloning described in this book allows you to roll your own Linux images without requiring such products. However, these products are more sophisticated than the simple clone script and z/VM configuration described in this book.

- Use of the `USER DIRECT` file versus a directory maintenance product

  The use of the `USER DIRECT` file was chosen over a directory maintenance product such as IBM DirMaint™ or CA's VM:Direct.

  If using DirMaint as a directory maintenance product would be better for your enterprise, you can refer to *Getting Started With Linux*, SC24-6096, to configure z/VM. To configure Linux, however, use this book.

- Use of predefined user IDs versus provisioning

  z/VM user IDs must be predefined to clone. There is no attempt to *provision* them (define and bring Linux user IDs online automatically) as part of the cloning process. The target Linux user ID must exist with the appropriate minidisks defined.

- Use of read-write versus shared read-only Linux `/usr/` file system

  Some cloning solutions use an environment that shares the `/usr/` file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. For this book, a read-write `/usr/` file system on the virtual servers was chosen to keep things as simple as possible.

- Use of conventional 3390 ECKD™ DASD versus FBA disks accessed through SCSI over FCP

  The System z server has traditionally only supported 3390 DASD. Support has been extended to include SCSI/FBA disks in storage area networks (SANs). The support of FBA disks is slightly more complicated than conventional DASD. In keeping things as simple as possible, only conventional DASD is described in this book.

Archived

- Use of manual installation versus cloning script or EXEC

  Two methods of cloning are described: manually, and through a Linux bash script. The manual method was chosen so that you will better learn the described concepts. However, the Linux script is also provided so you can save time.

1.4 Infrastructure design

To install and configure z/VM, and install, configure and clone Linux, or to provision virtual servers, there must be a certain infrastructure design in place. A System z server with associated resources and the z/VM operating system define much of this infrastructure.

Figure 1-1 shows a block diagram of a System z10 with multiple LPARs. z/VM 5.4 is installed in one of these LPARs. z/VM comes with many user IDs predefined. The most important six IDs are shown in the z/VM LPAR above the dashed line. Below the dashed line, you see the user IDs described in this book.

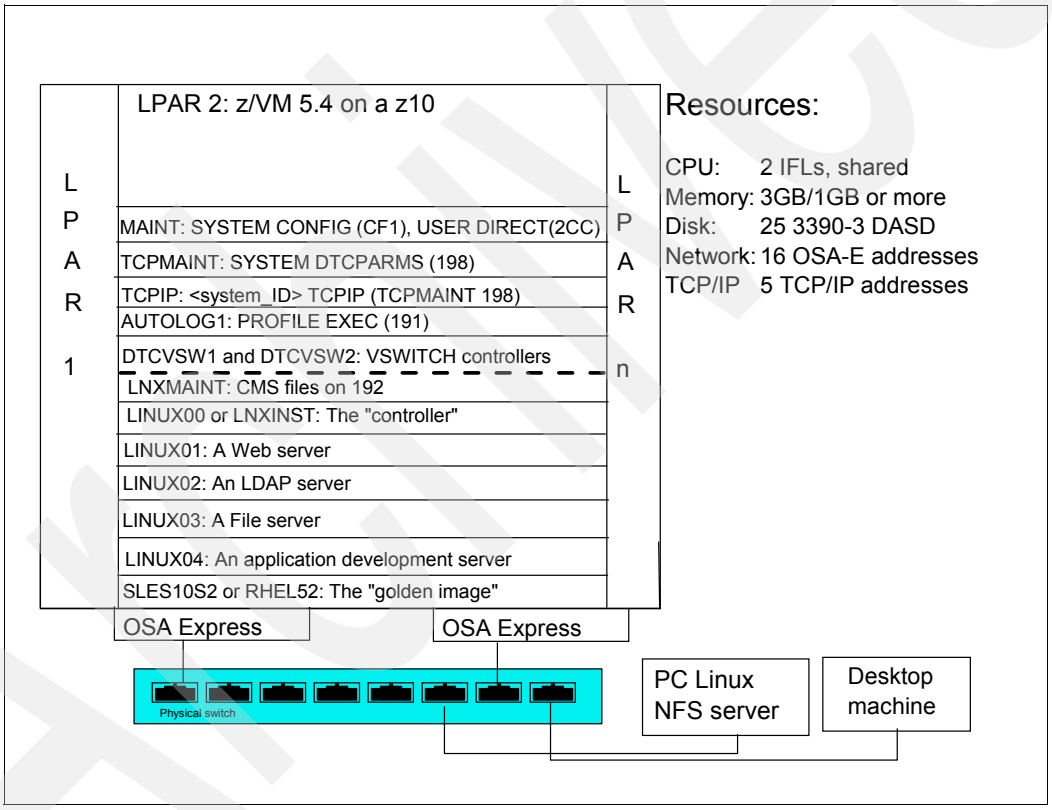


Figure 1-1 System infrastructure and z/VM user IDs

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

LNXMAINT This is a user ID on which to store files that will be used by both CMS and Linux.

LINUX00 or **LNXINST** This is the controller that does the cloning. It also serves as the Linux install server, and has other functions.

LINUX01-04 This is the user IDs that will be cloned to. Each virtual server is configured with a two 3390-3 minidisks to allow for slightly more than 4 GB of space.

SLES10S2 or **RHEL52** This is the SLES10 SP2 or RHEL 5.2 golden image. This is the Linux system that is cloned.

1.5 Usability tests performed for this book

During the writing of this book, many usability tests were conducted. The participants had a variety of skills, but none had both Linux and z/VM system administration skills.

By the end of the first day in all of the formal tests, most participants had all completed up to and including Chapter 5, “Servicing z/VM” on page 71, so z/VM was installed, serviced and customized for TCP/IP communications with a highly available VSWITCH.

By the end of the second day, most participants had cloned their first Linux virtual server. You should be able to complete most steps in the book in four solid days of work, if all goes well.

Archived

Archived

2

Planning

This chapter covers the planning needed before you install z/VM. It begins by discussing a *bill of materials*, or all the resources that you need. Then it describes conventions adopted for labeling 3390 volumes. Finally, it presents resource worksheets for:

- z/VM resources other than direct access storage device (DASD)
- DASD resources
- Linux resources
- Linux user IDs

Archived

© Copyright IBM Corp. 2008. All rights reserved.

2.1 Bill of materials

The resources needed for a Linux on System z project can be divided into:

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

- A System z or zSeries logical partition (LPAR); z800, z900, z890 or z990, System z9® or System z10
  - Processors or CPUs: One IFL (or CP) minimum; using two or more is strongly recommended
  - Memory: 3 GB central/1 GB expanded minimum; using 6 GB/2 GB or more is recommended. This 3:1 ratio of central to expanded storage is a useful starting point for relatively small systems such as in these example sizes.

    See the following Web site for a discussion about how to apportion memory:

    `http://www.vm.ibm.com/perf/tips/storconf.html`
  - DASD: 25 3390-3s or 9 3390-9s at a minimum
  - Open Systems Adapter (OSA) network cards: One card minimum with 8 device numbers (technically 6, but OSA “triplets” usually start on an even address). Two OSA Express cards with eight device numbers on one and four on the other is recommended for high availability.
- A network-attached computer that will act as an Network File System (NFS) server temporarily with at least 6 GB of disk space, but more may be needed. Setting up a Linux PC or UNIX® server is described.

  If you only have access to a Windows machine, AllegroNFS has been suggested as an NFS server; for information refer to

  `http://nfsforwindows.com/home`
- A workstation or desktop with network access to the mainframe

2.1.2 Software resources

The following software resources are needed:

- z/VM 5.4 install media with documentation (installation from DVD is described).
- SLES 10 SP2 (for the book SG24-7493) or RHEL 5.2 (for the book SG24-7492) Linux install media
- An operating system for the NFS server
- The code associated with this book
- Tools on the workstation and desktop:
  - A 3270 Emulator such as Attachmate Extra, Hummingbird Host Explorer, or IBM Personal Communications for Windows desktops
  - A Linux SSH client such as PuTTY (recommended) or TeraTerm
  - A VNC viewer

Archived

These resources are described in more detail in the chapters that follow.

2.1.3 Networking resources

The following network resources are needed:

- A TCP/IP address for z/VM
- One TCP/IP address for each Linux virtual server
- Associated TCP/IP information:
  - DNS host name
  - DNS domain
  - DNS server TCP/IP address
  - TCP/IP gateway
  - TCP/IP subnet mask
  - TCP/IP broadcast address (usually calculated from address and subnet mask)
  - TCP/IP MTU size

The TCP/IP addresses must be routed to the OSA cards.

2.2 z/VM conventions

It is good practice to use naming conventions so that you and others can recognize z/VM resources by their names. This section discusses such conventions for DASD volume names and backup file names.

2.2.1 Volume labeling convention

Use a convention for labeling DASD. Your enterprise may already have a labeling convention which will largely determine the labels to be given to the DASD used by your z/VM and Linux LPAR.

Each System z DASD is addressed with a device number consisting of four hexadecimal digits. Each System z 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, this convention guarantees that no two DASDs will have the same label. This can be an important issue, especially when z/OS® 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 that *owns* the DASD. The volume labeling convention used in this book identifies the LPAR through the first character. That leaves the second character in the label to identify the basic function of the DASD.

The LPAR used in this book is identified by the character M. 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)
S Spool space (SPOL)
T Temporary disk space (TDISK)
V z/VM operating system volumes

Archived

For example, Figure 2-1 shows the labeling convention for the DASD in LPAR *M*, of type *minidisk* at real address *A700*.

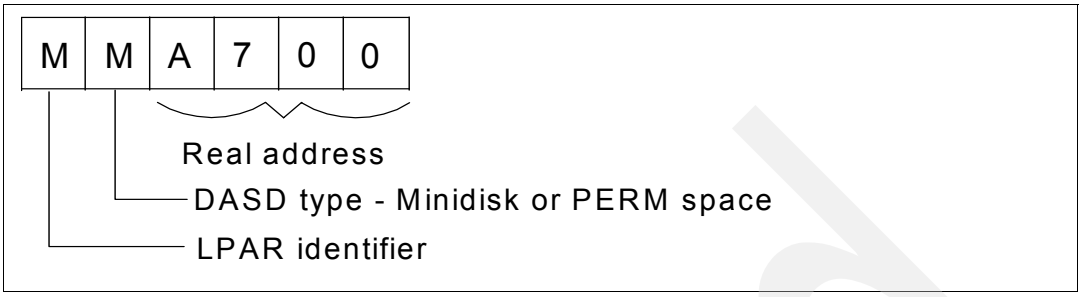


Figure 2-1 DASD labeling convention

The letter M is hard-coded into REXX EXECs that adopt this convention. If you want a different LPAR identifier character, they can easily be changed.

2.2.2 Backup file naming convention

It is recommended that you keep copies of important z/VM and Linux configuration files. You should 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. This often requires some characters to be overwritten. For the original file, the suffix `ORIG` is used, and for the most recent working copy, the suffix `WRKS` (for “it WoRKS”) is used. For example, the original `USER DIRECT` file is copied to the file `USER DIREORIG` before it is modified the first time.

2.2.3 The command retrieve convention

The ability to retrieve past commands is a common tool. Often it is useful to retrieve in both directions in case you “pass” the command that you are looking for. The default Linux shell, **`bash`**, does this by default with the up arrow and down arrow keys.

There is a convention in z/VM to use the F12 function key (labeled `PF12` on physical 3270 devices) to retrieve the last command, although it is not defined to all user IDs. There is no convention 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 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.

2.3 Disk planning

There are different aspects to consider when planning how to choose and allocate disk storage, including:

- Conventional ECKD DASD versus FBA disks over SCSI/FCP
- 3390-3s versus 3390-9s or large disk support
- Amount of disk storage per Linux image and how to allocate file systems

Archived

DASD versus SCSI/FCP

This book describes how to use conventional ECKD DASD and does not discuss FBA disks accessed over SCSI/FCP. This is not because either technology is superior, but simply because DASD seems to be much more common than SCSI/FCP disks. If you were to use SCSI/FCP disks, cloning through the **clone.sh** script would have to be modified to account for World Wide Port Names and Numbers.

Sometimes a combination of these two types of disk storage is used; in that case the ECKD emulated DASD is often used for the root file system and SCSI/FCP disks are used for large data storage areas.

3390-3s versus 3390-9s

Emulated 3390-3s format to about 2.3 GB. In contrast, 3390-9s are three times the size, or about 6.8 GB. Either size will work, although 3390-3s have been recommended over 3390-9s by some performance analysts.

This book describes mainly using 3390-3s; however, comments are added where using 3390-9s differs, especially with installing z/VM.

Disk storage per Linux image

Disk storage has the following characteristics

- This version of the book recommends using two 3390-3 DASD to create minidisks at virtual addresses 100 and 101. Previous versions only recommended a single minidisk at virtual address 100.
- The root file system is on `/dev/dasda1` with a recommended size of 384 MB. It is not a logical volume so that if there are any problems with LVM, the system will still be able to boot.
- Other file systems are on logical volumes that are part of a single volume group with the characteristics listed in Table 2-1.

Table 2-1 Recommended logical volume file systems and sizes

Mount point	Logical volume name	Size
`/usr/`	`usr-lv`	2 GB
`/var/`	`var-lv`	512 MB
`/opt/`	`opt-lv`	384 MB
`/tmp/`	`tmp-lv`	384 MB

This layout uses about 3.5 GB out of 4.5 GB of disk space. You could choose to use disk sizes other than 3338 cylinders (3390-3 minus cylinder 0). For example, if you chose to use 3390-9s, you could give 100 and 101 each half of the volume, giving each Linux about 6.8 GB of disk space.

Important: However you choose to lay out the minidisks, it is important that the golden image and all target Linux user IDs have two minidisks of the same size at virtual addresses 100 and 101. These assumptions are coded into the **clone.sh** script.

Archived

2.4 Memory planning

Planning memory may be the most difficult issue with z/VM and Linux on System z, but it is the most important to ensure adequate performance. The simplest solution may appear to involve having enough central memory (storage) in the LPAR so that z/VM never pages and Linux never swaps. However, realistically such resource is often not available.

A useful rule of thumb is to allocate memory on a “just enough” basis for each Linux server. A starting point is to set a virtual machine size by changing the memory allocation value at just over the value at which the guest starts to swap at the Linux system level when under normal loading. If some level of sustained swapping is inevitable due to the nature of the workloads, then ensure virtual disks are used for the swap media.

An understanding of memory planning is recommended, and resources that cover this important topic are listed here.

- IBM Redbooks publication *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926:

  http://www.redbooks.ibm.com/redpieces/abstracts/sg246926.html?Open

- IBM z/VM Performance Resource pages:

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

- The IBM z/VM page specifically discussing memory allocation:

  http://www.vm.ibm.com/perf/tips/storconf.html

One rule that can be recommended is to only have as few virtual machines logged on (or disconnected) as possible to handle the workload being presented. Every virtual machine that is not required should be logged off where appropriate, because this will mean more memory for the other virtual servers that remain running.

2.5 Password planning

Secure passwords are critical to reliable security. However, requiring many different passwords generally leads to people writing them down, which clearly detracts from security. Sometimes it is difficult to balance these two extremes, especially when dealing with system administration roles.

This book considers different system administration roles:

- The z/VM system administrator
- The Linux system administrator
- The Linux virtual server users

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

The method of backing up z/VM data onto the Linux controller 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: lnx4vm. If the z/VM and Linux system administrator roles must be kept separate and the Linux administrator is not to have access to the z/VM passwords, then a different method of backing up z/VM data must be chosen.

Archived

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

- The main z/VM system administrator (`MAINT`)
- The z/VM network administrator (`TCPMAINT`)
- The z/VM Linux administrator (`LNXMAINT`, Linux controller, Linux virtual server user IDs)
- The Linux virtual server users (with or without access to 3270 sessions, with or without the root passwords)

The sets of passwords that you define will depend on the roles that your organization will adopt.

2.6 Planning worksheets

Four worksheets are included in this section. They are populated with the resources used in writing this book. There are also four corresponding blank worksheets in 2.7, “Blank worksheets” on page 16, for your use.

2.6.1 z/VM resources used in this book

Table 2-2 lists the z/VM resource values used in the examples in this book. You can use these values as a reference for completing the blank worksheets that follow.

Table 2-2 z/VM resources worksheet

Name	Value	Comment
LPAR name	P21	3 GB main storage/1 GB expanded, 4 shared IFLs
CPC name	PELCP01	Name of CPC on which the LPAR is located
z/VM system name	VMLINUXA	Name to be assigned to z/VM system
TCP/IP host name	virtc522	Assigned by a network administrator; helpful to set in DNS beforehand, but not necessary
TCP/IP domain name	itso.ibm.com	Helpful to set in DNS beforehand
TCP/IP gateway	9.12.4.1	The router to and from the local subnet
DNS server 1	9.12.6.7	Assigned by the network administrator
DNS server 2/3 (optional)		Not used
OSA device name	eth0	Name of the interface to be assigned by IPWIZARD
OSA starting device number	3020	Start of OSA *triplet* for the z/VM TCP/IP stack
TCP/IP address	9.12.5.22	The TCP/IP address of the z/VM system
Subnet mask	255.255.255.0	Assigned by network administrator
OSA device type	QDIO	Often “QDIO” for OSA/Express cards
Network type	Ethernet	Usually “Ethernet”
Port name (optional)		Not required by z/VM
Router type	None	Usually “None”

Archived

Name	Value	Comment
MTU size	1500	Check with network administrator
Primary OSA device number for VSWITCH	3024	Specify the first device number (must be even number) and the next two device numbers will also be used
Secondary OSA device number for VSWITCH	3028	Should be on a different CHPID/OSA card

2.6.2 z/VM DASD used in this book

Table 2-3 lists the z/VM DASD resource values used in the examples in this book.

Table 2-3 z/VM DASD used in this book

Device number	Label	Type	Notes
D850	540RES	CP owned	z/VM system residence volume
D851	540SPL	CP owned	z/VM spool volume 1
D852	540PAG	CP owned	z/VM paging volume 1
D853	540W01	CP owned	z/VM first work volume
D854	540W02	CP owned	z/VM second work volume
D855	MPD855	CP owned	Paging volume 2
D856	MMD856	System minidisk	`LINUX00` or `LNXINST 100` (the controller)
D857	MMD857	System minidisk	`LNXMAINT` 191, 192 (common CMS files), `LINUX00` or `LNXINST 101` (used for `/nfs/` logical volume)
D950	MPD950	CP owned	Paging volume 3
D951	MPD951	CP owned	Paging volume 4
D952	MMD952	System minidisk	`LINUX00` or `LNXINST 102`
D953	MMD953	System minidisk	`LINUX00` or `LNXINST 103`
D954	MMD954	System minidisk	`LINUX00` or `LNXINST 104`
D955	MMD955	System minidisk	`SLES10S2` or `RHEL52 100` (the golden image)
D956	MMD956	System minidisk	`SLES10S2` or `RHEL52 101`
D957	MPD957	CP owned	`LINUX01 100`
DA50	MMDA50	System minidisk	Paging volume 5
DA51	MMDA51	System minidisk	`LINUX01 101`
DA52	MMDA52	System minidisk	`LINUX02 100`
DA53	MMDA53	System minidisk	`LINUX02 101`
DA54	MMDA54	System minidisk	`LINUX03 100`
DA55	MMDA55	System minidisk	`LINUX03 101`

Device number	Label	Type	Notes
DA56	MMDA56	System minidisk	LINUX04 100
DA57	MMDA57	System minidisk	LINUX04 101
DB50	MPDB50	CP owned	Paging volume 6

2.6.3 Linux resources used in this book

Table 2-4 lists the Linux PC NFS server resources used for the first System z Linux install:

Table 2-4 Linux NFS server resources used in this book

Name	Value	Comment
TCP/IP address	9.12.4.185	
User/password	root/lnx4vm	
NFS-exported install directory	/nfs/sles10sp2/ or /nfs/rhel52/	Directory with DVD 1

Table 2-5 lists the Linux resources used in the examples in this book.

Table 2-5 Linux resources used in this book

Name	Value	Comment
Linux install password	lnx4vm	
TCP/IP gateway	9.12.4.1	Obtain from network administrator
Subnet mask	255.255.252.0	Obtain from network administrator
DNS server	9.12.6.7	Obtain from network administrator
VNC installation password	lnx4vm	

2.6.4 Linux user IDs used in this book

Table 2-6 lists the z/VM user IDs for Linux used in the examples in this book.

Table 2-6 Linux user ID used in this book

Linux user ID	IP address	DNS name	Notes
SLES10S2 or RHEL52	9.12.5.29	virtc529.itso.ibm.com	SLES 10 SP2 or RHEL 5.2 golden image
LINUX00 or LNXINST	9.12.5.30	virtc530.itso.ibm.com	The controller
LINUX01	9.12.5.31	virtc531.itso.ibm.com	A Web virtual server
LINUX02	9.12.5.32	virtc532.itso.ibm.com	An LDAP virtual server
LINUX03	9.12.5.33	virtc533.itso.ibm.com	A file and print virtual server
LINUX04	9.12.5.34	virtc534.itso.ibm.com	An application development virtual server

Archived

2.7 Blank worksheets

Blank copies of the same four worksheets are provided for your use.

2.7.1 z/VM resources worksheet

Use the worksheet in Table 2-7 to document the z/VM resources that you will use.

Table 2-7 z/VM resources blank worksheet

Name	Value	Comment
LPAR name		
CPC name		
System name		
TCP/IP host name		
TCP/IP domain name		
TCP/IP gateway		
DNS server 1		
DNS server 2/3 (optional)		
OSA device name		Often “eth0”
OSA starting device number		
TCP/IP address		
Subnet mask		
OSA device type		Often “QDIO”
Network Type		Often “Ethernet
Port name (optional)		
Router Type		Often “None”
Primary OSA device number for VSWITCH		
Secondary OSA device number for VSWITCH		Should be on a different CHPID/OSA card than primary

Archived

2.7.2 z/VM DASD worksheet

Use the worksheet in Table 2-8 to document the z/VM DASD that you will use.

Table 2-8 z/VM DASD blank worksheet

Device number	Label	Type	Notes

Archived

2.7.3 Linux resources worksheet

Use the worksheet in Table 2-10 to document the resources associated with the NFS server that will be used to be the install source of the first System z Linux.

Table 2-9 Linux NFS server resources blank worksheet

Name	Value	Comment
TCP/IP address		
User/password		
NFS-exported install directory		

Use the worksheet in Table 2-11 to document your System z Linux resources.

Table 2-10 Linux resources blank worksheet

Name	Value	Comment
Linux install password		
Linux root password		
Apache user ID and password		
Linux TCP/IP gateway		
Linux TCP/IP broadcast		
Linux DNS server		
VNC Installation password		

2.7.4 Linux user ID worksheet

Use the worksheet in Table 2-11 to document the Linux user IDs that you will create.

Table 2-11 Linux user ID blank worksheet

Linux user ID	IP address	DNS name	Notes

Archive

3

Configuring a desktop machine

Many people use Microsoft® Windows as a desktop operating system. This chapter addresses the use of the following tools, which are recommended for accessing z/VM and Linux from a Windows desktop:

- An SSH client: PuTTY is recommended
- A VNC client: RealVNC is recommended
- A 3270 emulator: Many choices are available

Archived

© Copyright IBM Corp. 2008. All rights reserved.

3.1 PuTTY: a free SSH client for Windows

Throughout this book, SSH is used to log into Linux systems. It is easy to use and cryptographically secure. If you are using a Linux desktop system, an SSH client is built in. But if you are using a Windows desktop, you will need a useful SSH client.

PuTTY is probably the most commonly used SSH client. You can find a PuTTY client for Windows on CD1 of a SLES 10 distribution in the `/dosutils/putty` directory. You can download PuTTY from the Web at:

`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 standalone executable (no installation is needed, other than copying the file). You may also want to create a shortcut on your desktop or task bar.

Open PuTTY and the configuration window shown in Figure 3-4 on page 22 should open. If you spend a few minutes at this point configuring PuTTY, it may pay off in a time savings later. The examples shown use PuTTY Release 0.60.

1. In the PuTTY Configuration window, in the left Category panel, click **Session**.
2. Under the Connection Type heading on the top right, click the **SSH** radio button as shown in Figure 3-1. This specifies the use of the SSH protocol.

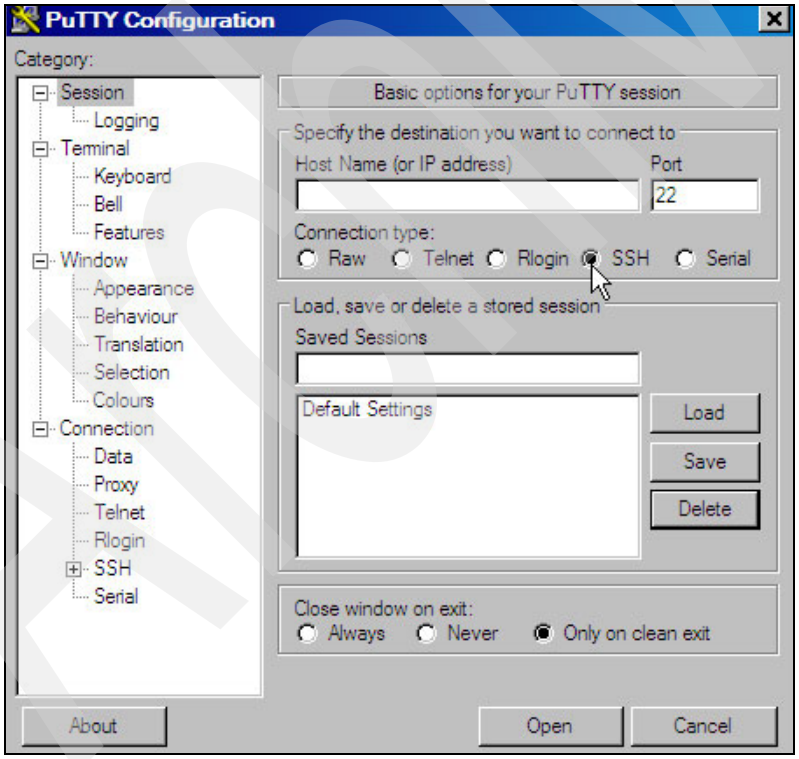


Figure 3-1 PuTTY Configuration window

3. Click **Logging** in the left panel as shown in Figure 3-2 on page 21.
   - Click the radio button **Printable output** in the Session logging radio group. This will allow you to go back and check the output of certain commands.
   - Set the Log file name to `&H&M&D&T.log` so a time stamp will be in the file name.

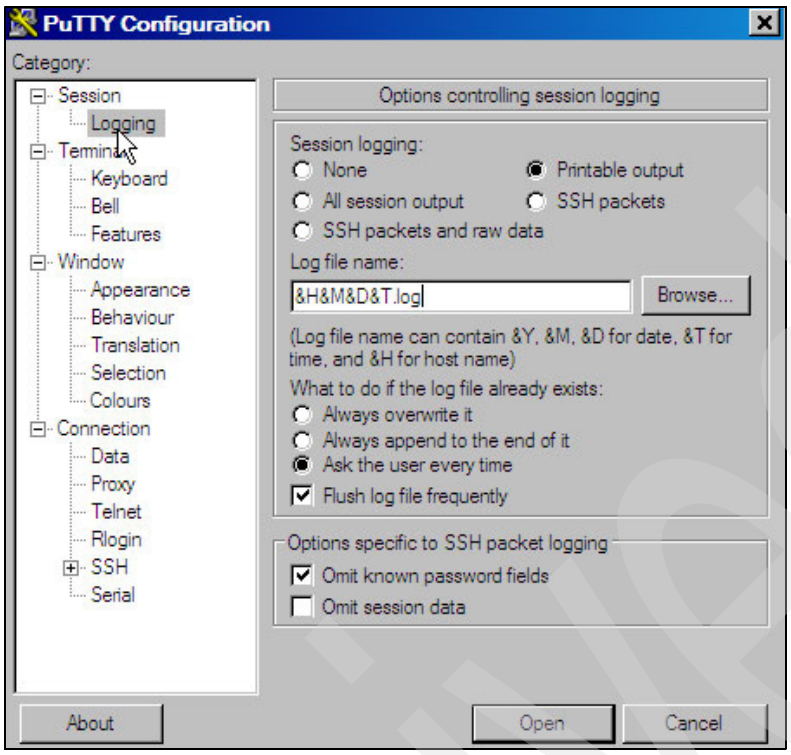


Figure 3-2 Setting logging

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

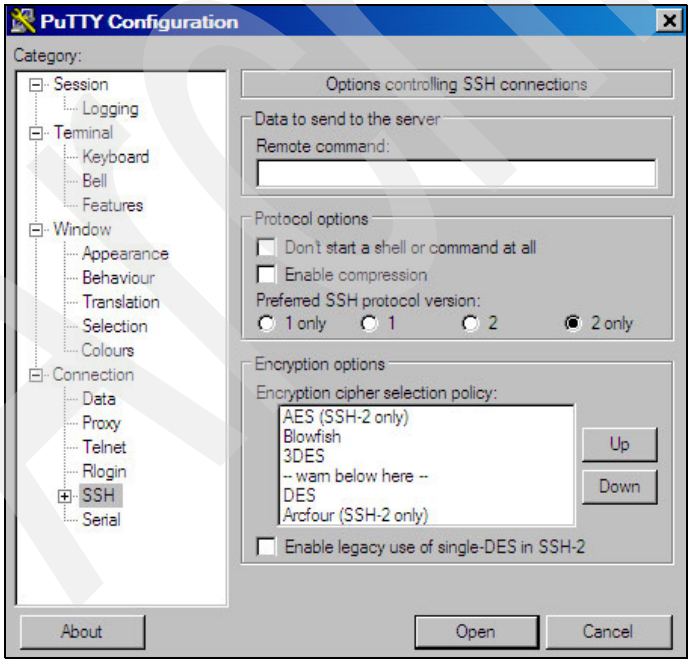


Figure 3-3 Setting SSH Protocol 2

6. In the left Category panel, click **Terminal** as shown in Figure 3-4 on page 22.

7. Select the **Use background colour to erase screen** check box, which results in a better job of painting the panel for applications that use curses (block graphics).

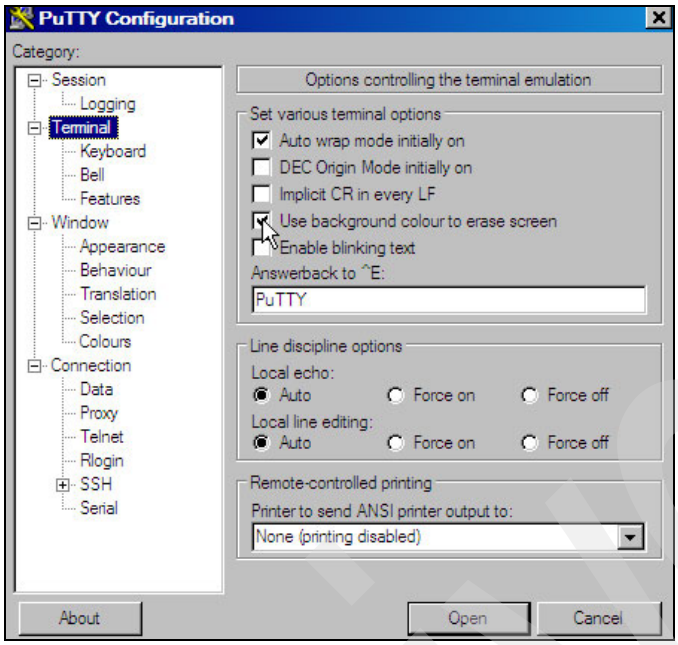


Figure 3-4 Customizing PuTTY SSH settings (Part 1 of 4)

8. Click **Window** in the left pane, as shown in Figure 3-5.
9. You may choose a larger window size and more lines of scrollback. In this example, 50 rows, 100 columns are and 1000 lines of scrollback are set.

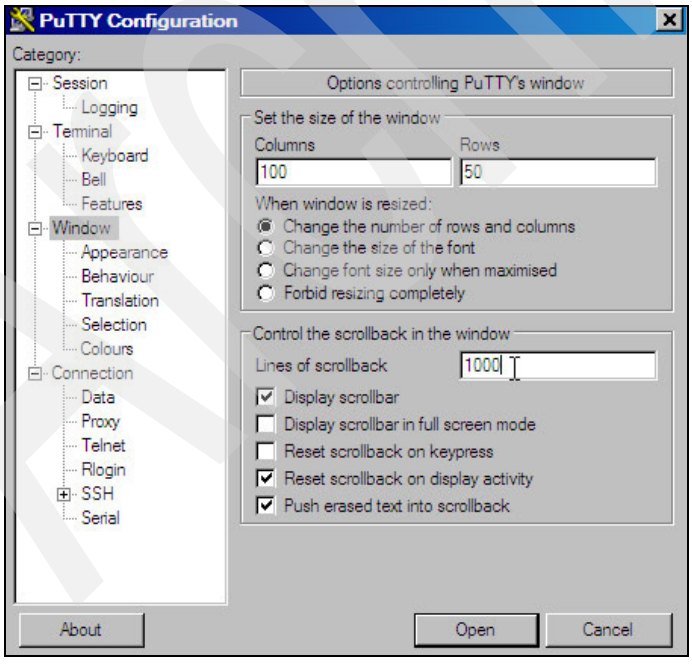


Figure 3-5 Setting Window and scrollback size

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

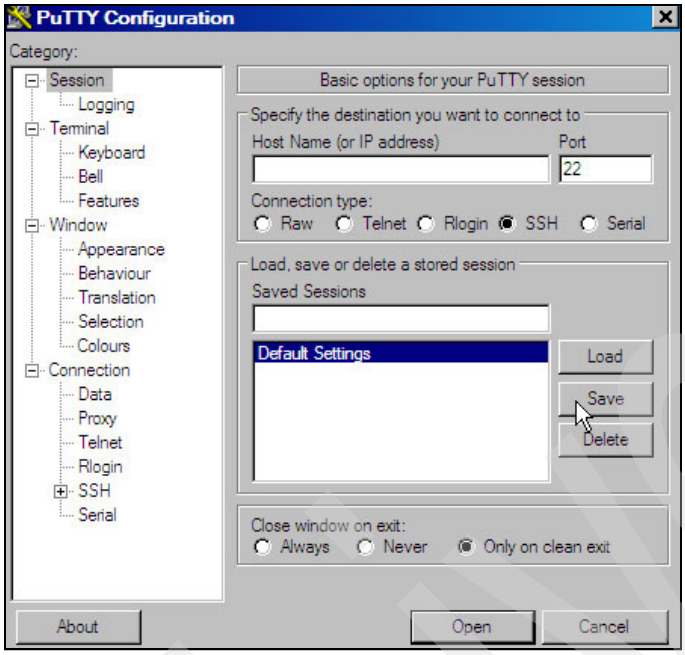


Figure 3-6 Saving new default settings

Saving sessions

To save sessions, perform the following steps. In this example a session for `LINUX00`, or the controller, is saved.

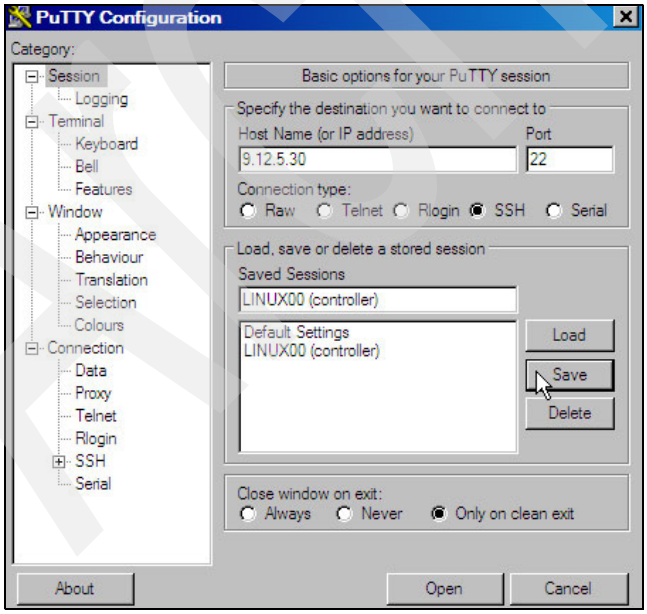


Figure 3-7 Customizing PuTTY window settings (part 4 of 4)

To save a session for each virtual server, perform the following:

1. In the Host Name (or IP address) field, enter the TCP/IP address (or 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 should see the name added to the Saved Session list.

   Now whenever you start PuTTY, you can simply double-click any saved session name, and an SSH session to the desired Linux system will be invoked.

3.2 Setting up a VNC client

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

If you are using a Linux desktop you probably have, or at least have access to, a VNC client named **`vncviewer`**. It is part of the tightvnc package.

3.2.1 Downloading and running RealVNC

If you have a Windows desktop, the VNC client from RealVNC is a popular choice (you can also find a TightVNC client for Windows on CD1 of a SLES 10 distribution in the `/dosutils/tightvnc` directory).

You can purchase a full function RealVNC client, or there is a free version. The RealVNC home page is:

`http://www.realvnc.com`

The download page is:

`http://www.realvnc.com/download.html`

Fill out the Web form and download the executable. When you have downloaded it, run it and an install program will start. At the time of writing, RealVNC 4.1.2 was the current version.

Accept all defaults, however, you probably do not need a VNC server on your desktop. So you can **deselect VNC Server** from the *Select Components* window as shown in Figure 3-8.

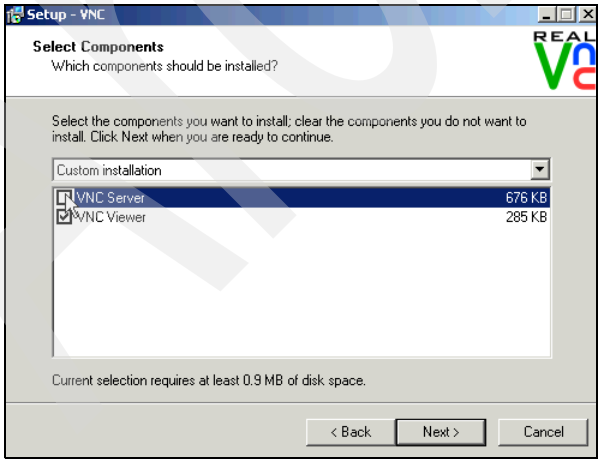


Figure 3-8 RealVNC Select Components window

Complete the screens and the installation process should go quickly.

3.2.2 Customizing RealVNC

The latest VNC protocol is Version 4, which is the default with the VNC client. This version will work with the VNC servers shipped with SLES 10 or RHEL5.

If, however, you need to use protocol Version 3.3 for SLES 9, open the VNC client and click the **Options** button, as shown in the left side of Figure 3-9. Click the **Misc** tab. Click the check box **Use only protocol version 3.3** as shown in the center of the figure. Finally, click the **Load/Save** tab and click **Save** to save the changes.

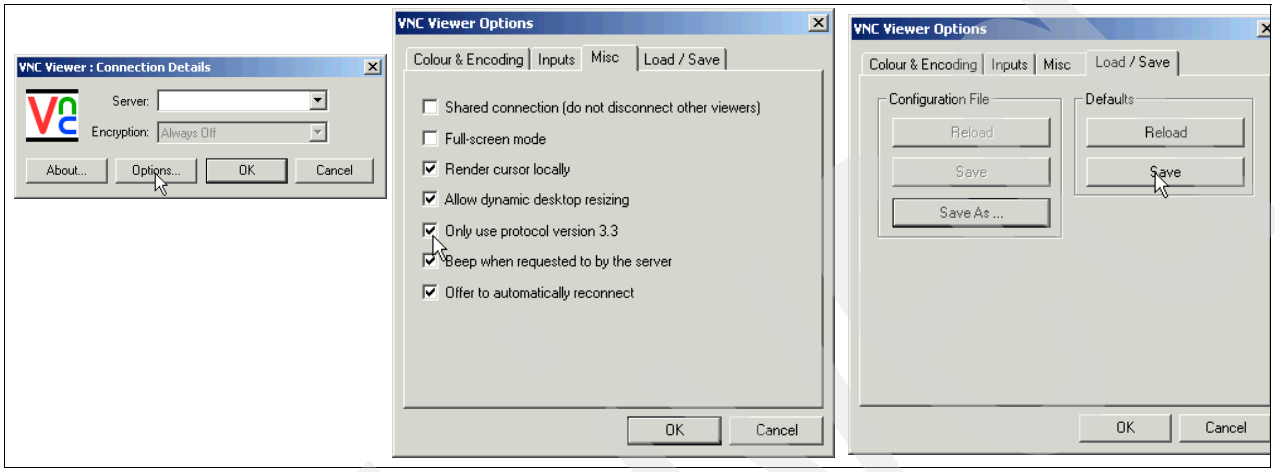


Figure 3-9 Setting VNC client to use protocol 3.3 - required for SLES 9 only

Your VNC client should now be ready to connect to the VNC server that your Linux systems will have.

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. Some of the more common ones are:

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

It is beyond the scope of this book to explain the details of configuring all the various emulators. However, it is recommended that you investigate the following settings for your emulator:

- Set the Enter and Clear function keys to be where you would expect them.

  On some emulators, the default Enter key action is set to the right Ctrl key of modern keyboards. Likewise, the Clear key action is sometimes set to the Esc key in the upper left corner of modern keyboards, or to 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 32, 43, or more lines if they can easily fit in a window, given your desktop display size and resolution.

- Have the session automatically reconnect after logoff.

  Having a new logon window come back immediately after you log off can also save you time in the long run. This is often not the default behavior.

- Save your connection sessions.

  Rather than continually typing in the IP address or DNS name of the z/VM system to which you want to connect, spend a few minutes defining and saving a session for each system to which you may connect, as described for PuTTY. Then you can usually double-click the saved connection to quickly access a new 3270 session.

Archived

4

Installing and configuring z/VM

To complete this chapter, you must complete the majority of Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85.

However, we recommend that you start here, because the **instdvd** step (used when installing z/VM) listed in 4.1.2, “Copying a vanilla z/VM system to DASD” on page 31, takes two or more hours to complete. While that process is running, you can configure the Network File System (NFS) server. Alternatively, if you have other personnel who can work on the project, you can start both chapters at the same time on the different systems.

In this chapter, you perform the following steps:

- Installing z/VM from DVD
- Configuring TCP/IP
- Configuring the XEDIT profile
- Customizing the SYSTEM CONFIG file
- Configuring TCP/IP to start at IPL time
- Adding paging volumes
- Creating a user ID for common files
- Addressing z/VM security issues
- Relabeling the system volumes

© Copyright IBM Corp. 2008. All rights reserved.

4.1 Installing z/VM from DVD

The section that follows assumes a first-level installation of z/VM from DVD onto DASD. If you have not already done so, complete the worksheet that is provided in 2.7.1, “z/VM resources worksheet” on page 16. Note the following points:

- For System z9 hardware and older, you will need access to the Hardware Management Console (HMC) with a user ID that has authority to go into single object operations mode.
- For System z10 hardware and newer, the requirement to be in single object operations mode to access the Load from CD-ROM, DVD or Server function has been removed.

z/VM 5.4 is shipped on tape and DVD. z/VM should install faster from tape due to faster I/O speeds; however, installing from tape may require more trips between and the HMC and the tape drive.

- If you are familiar with the use of the HMC, you can use the two-page *z/VM Summary for Automated Installation and Service (DVD Installation)* to replace or augment the description that follows.
- If you are not familiar with the HMC and z/ VM, you may want to use the complete installation manual *z/VM Guide for Automated Installation and Service Version 5 Release 4.0*, GC24-6099.
- If you are installing z/VM at the *second level* (that is, z/VM under z/VM) or onto SCSI disk, use the GC24-6099 publication because the sections that follow do not address these options.

4.1.1 Booting z/VM from DVD

This section explains how to install z/VM 5.4 from an HMC with a DVD-ROM onto 3390-3 equivalent DASD. Some words are included for installing onto the larger 3390-9 DASD. For alternative configurations, such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

1. Logon to the Hardware Management Console. You should see the HMC Workplace™ window.
2. Select the LPAR on which you want to install z/VM; this is often reachable by clicking the **CPC images** icon.

   Note: Ensure that you have the *correct* LPAR selected. If you are not completely sure, check with your system administrator.

3. If necessary, click the **racetrack** buttons (two buttons that are circular arrows on the bottom right corner) to traverse to the Recovery or CPC Recovery menu.
4. On the Recovery or CPC Recovery menu, double-click the **Integrated 3270 Console**, as shown at the bottom of Figure 4-1 on page 29.

Archived

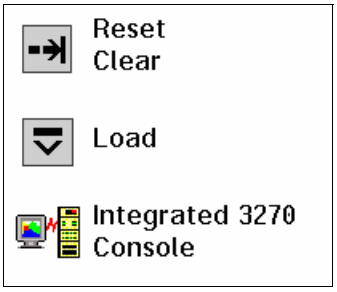


Figure 4-1 Integrated 3270 Console icon

A window entitled Integrated 3270 Console for <your CPC> will open. (On earlier HMC levels, the window may be entitled Personal Communications).

Hint: It is convenient to use the Alt-Tab key sequence to move between the HMC window and the 3270 console.

5. Insert the z/VM Product Package Version 5 Release 4.0 DVD into the HMC DVD drive.

Important: On z10 HMCs and later, it is no longer required to be in Single Object Operations mode in order to install z/VM.

6. On a z9 HMC and earlier model, get into Single Object Operations mode (as mentioned, for z10 HMCs and later, this step is not necessary). To get into this mode, perform the following steps:
   a. Double-click the **Groups** icon in the Views Area.
   b. Double-click **Defined CPCs** in the Groups Work Area.
   c. Select your CPC.
   d. If necessary, go around the **racetrack** (the buttons with circular arrows on the bottom right corner) to the CPC Recovery menu.
   e. Double-click the **Single Object Operations** icon. Click **yes** to confirm. A new window Primary Support Element Workplace should appear (on older HMC levels, it will be a “window within a window”). A window about a certificate not being valid may appear; in this case, press **OK**.
   f. Double-click **Groups** near the top of this window.
   g. Double-click **Images** in the Groups Work Area.

Important: If you are unable to get into Single Object Operations mode, it may be because you do not have sufficient permission. Check with the system administrator.

7. The LPAR that z/VM will be installed into should still be selected. On the right you should still see the (CPC) Recovery menu. Double-click the **Load from CD-ROM or Server** icon as shown in Figure 4-2 on page 30.

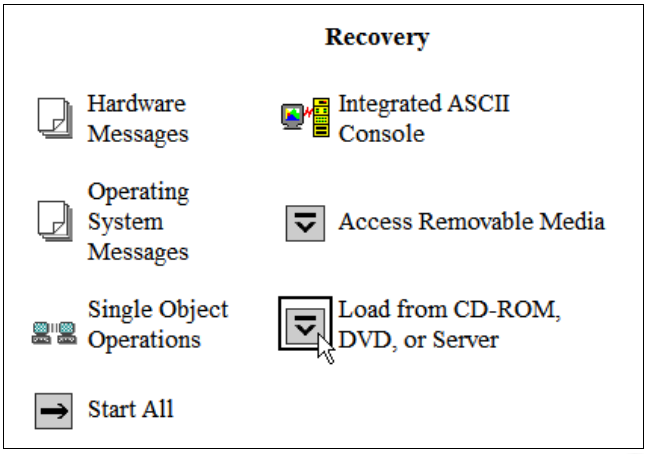


Figure 4-2 CPC Recovery menu with Load from CD-ROM or Server icon present

8. On the Load CD-ROM or Server window as shown in Figure 4-3, the radio button **Hardware Management Console CD-ROM / DVD** should be selected.
9. In the same Load CD-ROM or Server window, fill in the File location field with `/cpdvd`. This is the directory on the DVD with the z/VM 5.4 installation code.
10. Click **OK**.

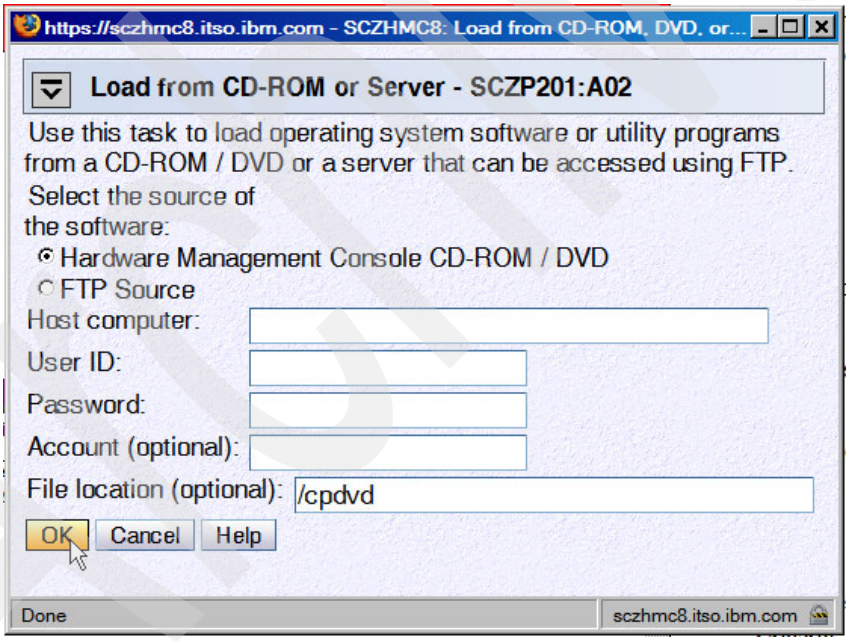


Figure 4-3 Load from CD-ROM or Server window

11. Load the RAMDISK:
    a. From the Load from CD-ROM or Server window, the software **540vm.ins** should be selected as shown in Figure 4-4 on page 31. Click **OK**.

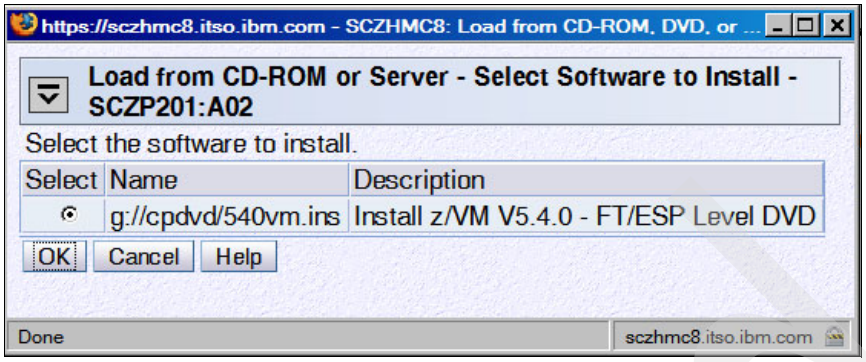


Figure 4-4 Selecting z/VM 5.4 RAMdisk system

b. From the Confirm the action window, click **Yes**. You should see the Load from CD-ROM, DVD or Server Progress window. The green light on the DVD drive should light up.

c. When you see the message `Completed successfully.`, click **OK** to close. This should normally take about two to four minutes.

> **Important:** Normally, the z/VM RAMdisk (IBMVMRAM) loads in about four minutes. However, slow load times have been observed (15 to 18 minutes). When the green light on the DVD drive is solid, the load time will be acceptable. When it is intermittently dark more than it is green, long load times can result. This will also balloon the estimated run time of 2 to 2.5 hours of the **`INSTDVD`** process, which may consume 5 to 6 hours. PMR 58909,180 was opened to address this issue.

At this point, you should now have an in-memory z/VM 5.4 system running.

4.1.2 Copying a vanilla z/VM system to DASD

This section describes the steps to copy z/VM to DASD.

1. You can now get out of Single object operations mode (if you are in it). To do so, log off the primary SE window by closing that window.
2. Move to the Integrated 3270 Console window (you can use the **Alt-Tab** sequence). The RAMdisk should IPL and you should see z/VM boot as shown in Figure 4-5 on page 32.

> **Note:** The Esc key clears the window on the HMC.

Archive

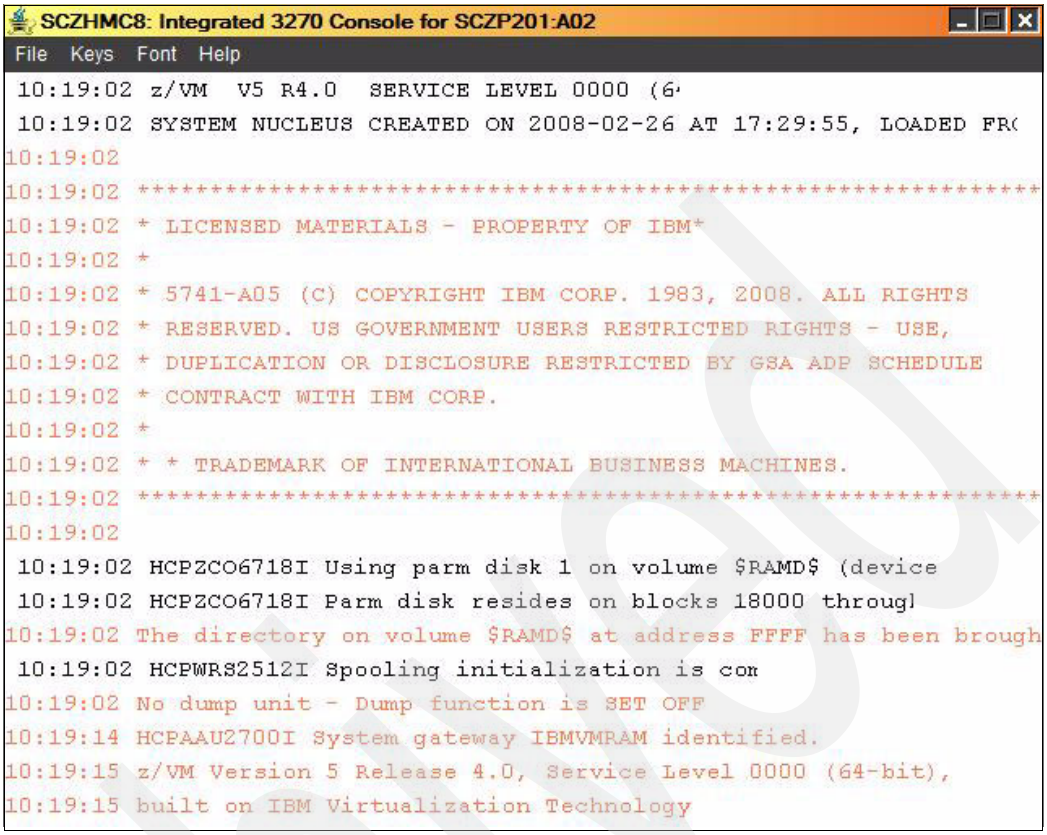


Figure 4-5 z/VM first boot on Integrated console

3. Invoke the **instplan** command. This will allow you to choose associated z/VM products to install, as well as the language to use and the type of DASD on which to install:

   ```
   ==> instplan
   ```

4. You should see the Installation Planning window display shown in Figure 4-6 on page 33. We recommend that you leave the Ms in the top section as is.

Archived

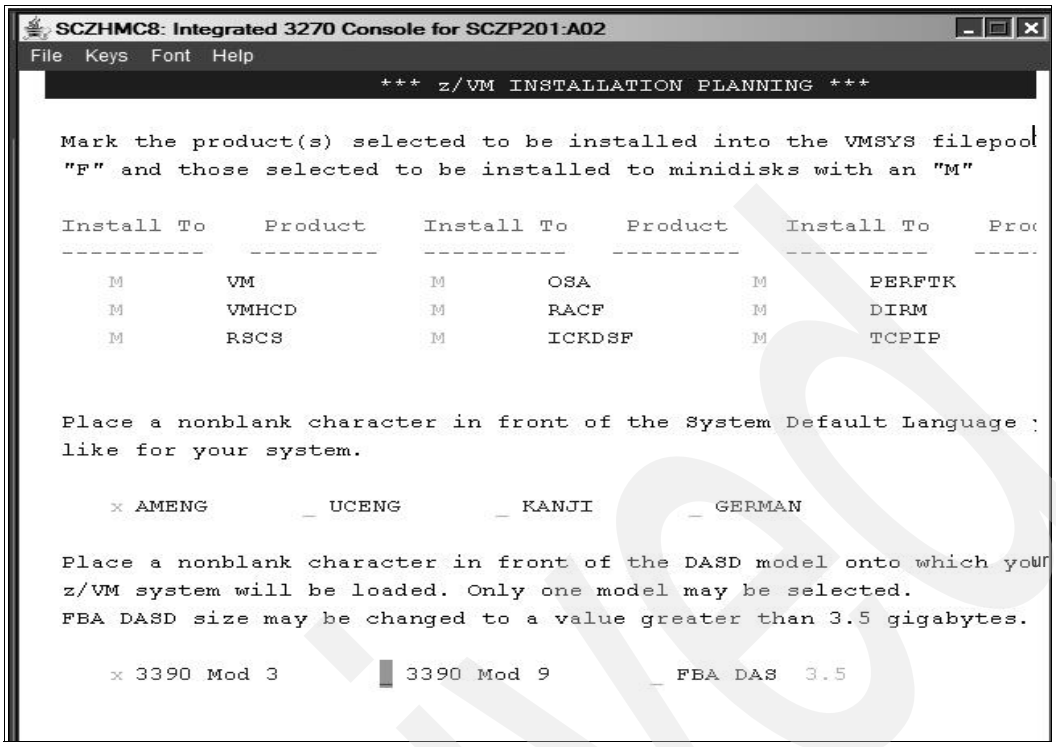


Figure 4-6 Installation Planning window

5. On this window, type `X` next to `AMENG` (or select your language) and type `3390 Mod 3` (or the type of DASD you will use), as shown in Figure 4-6.
6. Press **F5**. You should the message `HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY`.
7. Attach the DASD devices onto which z/VM will be installed as defined in your planning worksheet in 2.7.2, "z/VM DASD worksheet" on page 17. In this example, the devices are `d850-d854`.

```
==> att <d850-d854> *
d850-d854 ATTACHED TO MAINT
```

Important: The angle brackets <> in the example should *not* be typed. They are used throughout the book to signify that you should replace the example value with the correct value for your site. For example, if you are installing z/VM onto DASD 1200-1204, you would type the following:

```
==> att 1200-1204 *
```

Running INSTDVD

The INSTDVD EXEC copies the z/VM system from DVD to disk.

1. Execute the **`INSTDVD EXEC`**:

```
==> instdvd
```

2. If you are using 3390-3s, you see a window asking for the five volumes as shown in Figure 4-7 (if you are using 3390-9s, you will only see three lines).

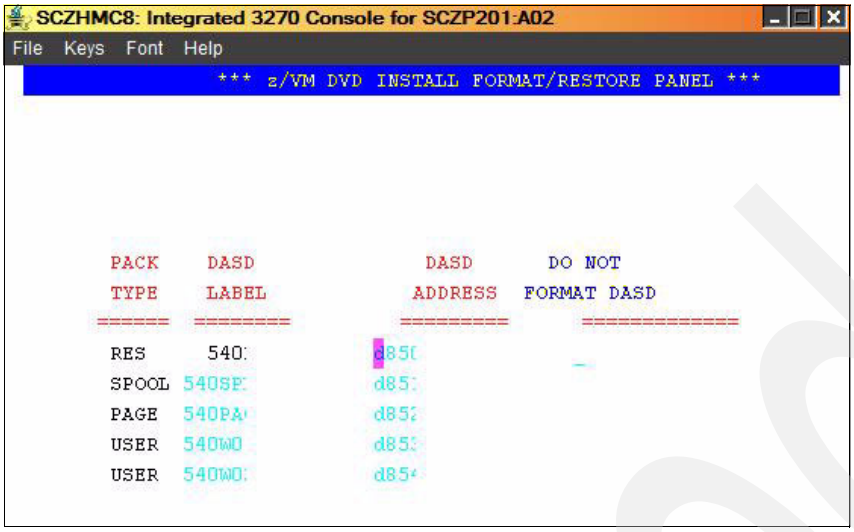


Figure 4-7 INSTDVD DASD address window

a. Enter the addresses of the five volumes (or three volumes, for 3390-9s) that z/VM will be installed on.

b. Press **F5** to start the installation.

3. Verify that the five DASD addresses to be installed onto are correct. When you see the question `DO YOU WANT TO CONTINUE?`, type **`Y`**. You should see the message `NOW FORMATTING DASD <D850>`.

> **Important:** INSTDVD can take from 45 minutes to two hours or more. You may want to use this time to set up an NFS server, as explained in Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85.
>
> Also, read errors have been observed resulting in **`INSTDVD`** failing. If this is the case, you can try the command **`instdvd (restart`** and the install process should pick up where the read error occurred. This error can be caused by dirt or fingerprints on the DVD.

4. You are asked to place the system RSU in the drive. Insert the z/VM Stacked Recommended Service Upgrade 5401 DVD into the HMC DVD-ROM drive.

5. At the Integrated 3270 Console, type `GO`. You should see a messages of the form `DVDLOAD: LOADING FILE CKD5000x IMAGE *.` This step should take two to four minutes.

6. Finally, you should see the message `HCPIDV8329I INSTDVD EXEC ENDED SUCCESSFULLY`.

4.1.3 IPL the vanilla z/VM from DASD

IPL your initial z/VM system now on DASD. Your 3270 Integrated Console session should still be running.

1. From the HMC Workplace window, **select your LPAR** by clicking it. You may have to first double-click **Groups**.

2. You should see the CPC Recovery (sometimes entitled simply Recovery) menu. Double-click the **Load** icon in the menu at the right side.

Archived

3. The Load window opens as shown in Figure 4-8. Follow these steps:
   a. Set the load address to the new system residence (540RES) volume, which is `D850` in this example.
   b. Set the load parameter to `SYSG`. This specifies the use of the Integrated 3270 console.
   c. Click **OK** to IPL.

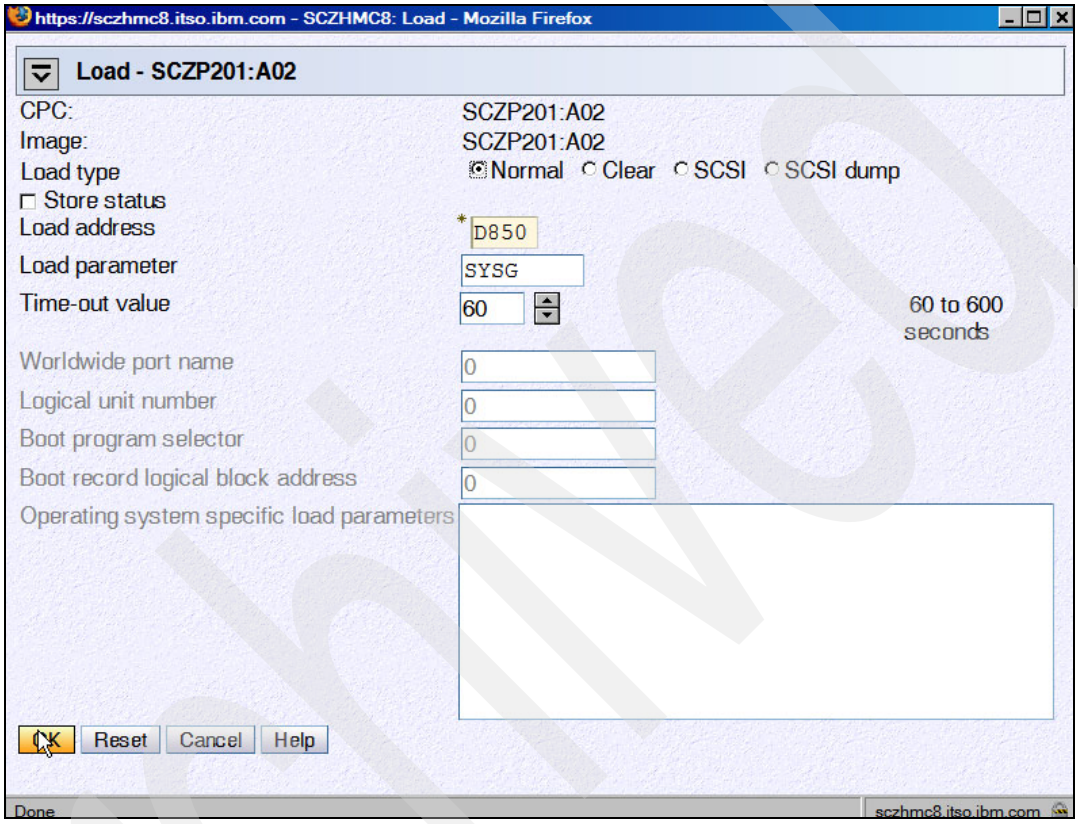


Figure 4-8 Load window

4. When you see the Load Task Confirmation window, click **Yes**.
5. After 1 to 3 minutes, you should see `Success` in the Load Program window. Click **OK**.
6. Move back to the Integrated 3270 console window. You should see the Standalone Program Loader window as shown in Figure 4-9 on page 36.
   a. Press the **Tab** key to traverse to the IPL Parameters section and enter the value **`cons=sysg`**. This specifies the use of the Integrated 3270 console.

Archive

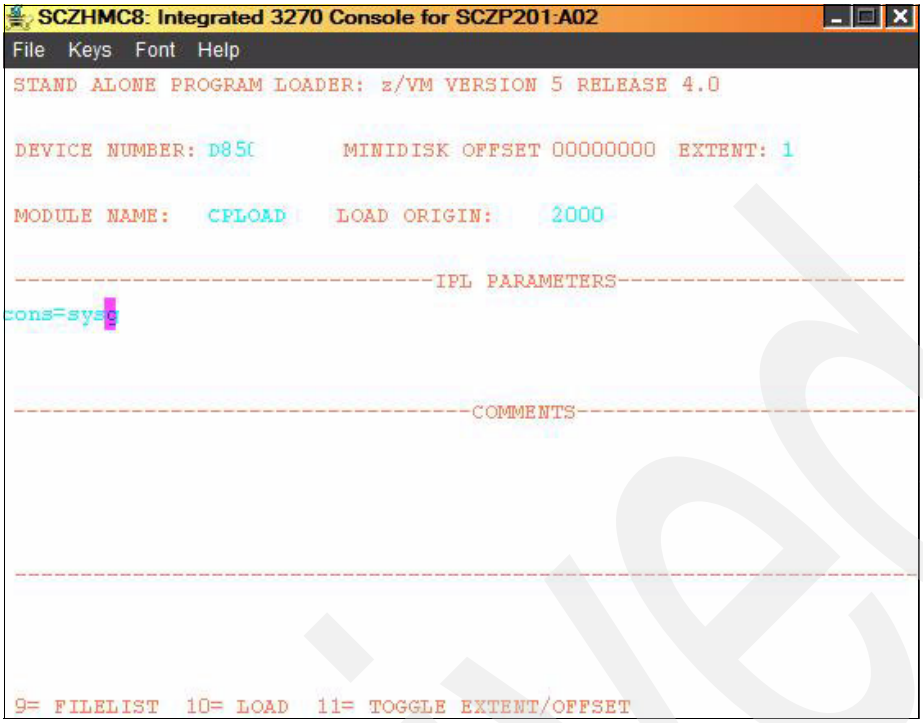


Figure 4-9 The Standalone Program Loader window

b. Press the **F10** key to continue the IPL of your z/VM system. This should take about 1 to 3 minutes.

7. At the `Start (Warm|Force|COLD|CLEAN)` prompt, enter:

```
==> cold drain noautolog
```

8. At the Change TOD clock prompt, enter:

```
==> no
```

9. The last message should be `HCPCRC8082I EREP records are accumulating for userID EREP`. Disconnect from the `OPERATOR` user ID through the **`DISCONNECT`** command:

```
==> disc
```

Press **Enter** to get a new logon window.

4.1.4 Completing the z/VM installation

Follow these steps to complete the z/VM installation

1. On the z/VM login window, logon as MAINT. The password is MAINT. You may receive messages `HCPLNM102E` or `HCPLNM101E` about disks not linked or attached. This is not a problem. Press **Enter** when you see the `VM Read` prompt in the lower right corner.

Important: When logging onto a z/VM user ID that runs CMS, you should usually press **Enter** at the `VM READ` prompt. This will result in a prompt of the form:

```
Ready; T=0.01/0.01 11:14:20
```

Archived

2. IPL CMS, then press **Enter** at the `VM READ` prompt in the lower right corner. You should see the `Ready;` prompt.

```
==> ipl cms
==> Press Enter at the VM READ prompt
```

3. Run the **instvm dvd** command:

```
==> instvm dvd
...
HCPPLD8329I POSTLOAD EXEC ENDED SUCCESSFULLY
...
HCPIVM8392I INSTVM ENDED SUCCESSFULLY
```

   This EXEC continues the installation process. This step should take about 4 to 8 minutes. The last message should be `HCPIVM8392I INSTVM ENDED SUCCESSFULLY`.

4. Load the recommended service. First **IPL CMS**, then press **Enter** at the VM READ prompt:

```
==> ipl cms
==> Press Enter at the VM READ prompt
Ready;
```

5. For z/VM 5.4, the service name is **5401RSU1**. Verify this file exists on the `MAINT 500` disk:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * * c
5401RSU1 SERVLINK C1
```

6. Run the **SERVICE ALL** command to apply the service:

```
==> service all 5401rsu1
```

   This step should take about 3 to 6 minutes. The last message should be:

```
VMFSRV2760I SERVICE processing completed successfully.
```

7. Now IPL CMS and run the **put2prod** command. This puts the service into production:

```
==> ipl cms
==> Press Enter
Ready;
==> put2prod
```

   This step should take about 2 to 4 minutes. The last message should be:

```
VMFP2P2760I PUT2PROD processing completed successfully.
```

   A return code of zero (0) is ideal. You may get a return code of 4 and the message:

```
VMFP2P2760I PUT2PROD process completed with warnings.
```

   In general on z/VM, a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

8. Enter the following command to shut down and re-IPL your system:

```
==> shutdown reipl
SYSTEM SHUTDOWN STARTED
```

9. You will lose the current session on the Integrated 3270 Console, but the system should come back in about 2 to 4 minutes.

10. After it comes back, the last message should be `Press enter or clear key to continue`. Press **Enter** and you should see a z/VM logon window.

At this point, you should now have a vanilla z/VM system installed.

Archived

4.2 Configuring TCP/IP

We recommend that you initially configure TCP/IP through the **IPWIZARD** command, which is generally used just once. After **IPWIZARD** creates the initial configuration files, they are typically maintained manually.

From the z/VM logon window, logon to MAINT. The default password for all z/VM user IDs is the same as the user ID. So enter a password of **maint**, which will not be echoed on the window.

```
USERID   ==> maint
PASSWORD ==>
```

After entering the user ID and password, press **Enter** when the status area in the lower right shows `VM READ`.

4.2.1 Use the IPWIZARD tool

The **IPWIZARD** command is on the `MAINT 193` disk. You will need to access it as file mode G through the **ACCESS** command so that you will pick up **IPWIZARD** from that minidisk.

1. Access the `MAINT 193` disk:

   ```
   ==> acc 193 g
   ```

2. Invoke **IPWIZARD**.

   ```
   ==> ipwizard
   ```

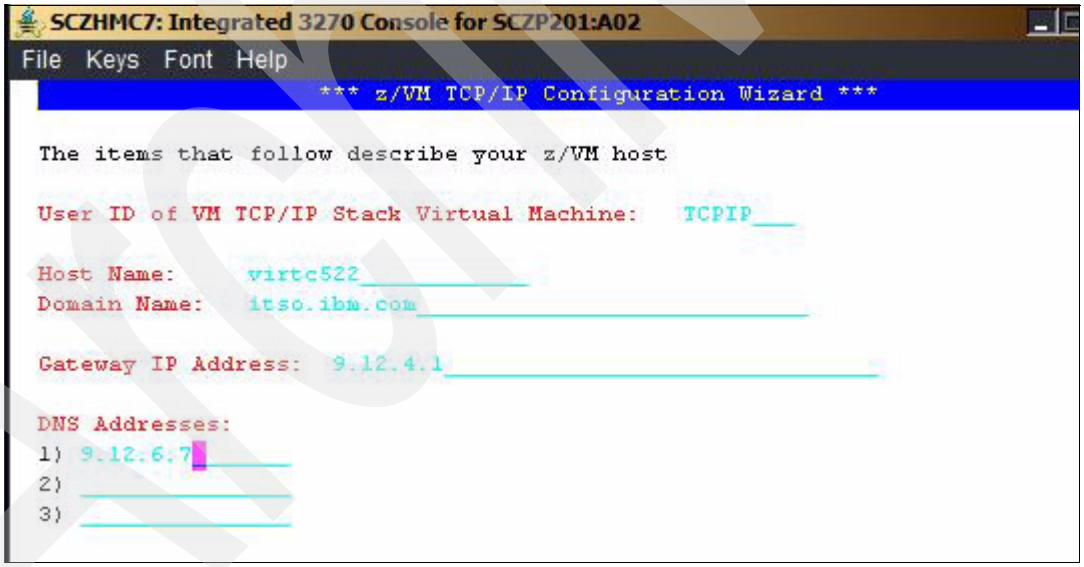


Figure 4-10 IPWIZARD window 1

3. The z/VM TCP/IP Configuration Wizard opens as shown in Figure 4-10. The first field, User ID, should always be `TCPIP`. Obtain the remaining values from the 2.7.1, “z/VM resources worksheet” on page 16 and press **F8**.

Archived

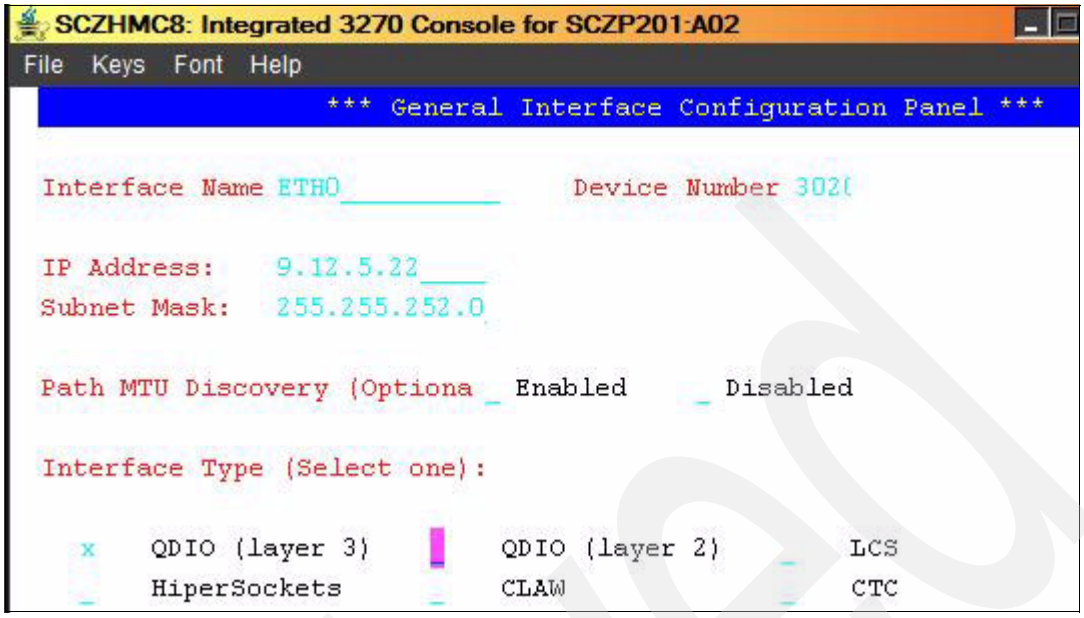


Figure 4-11 IPWIZARD window 2

4. An Interface Name of `ETH0` is arbitrary but recommended. The Device Number will be the starting address of the OSA triplet that the z/VM stack will use. The IP address that must be routed to the OSA card will become the TCP/IP address of the z/VM system. The Interface Type will typically be `QDIO (layer 3)` with modern OSA devices. When you finish, press **F8**.

> **Note:** To utilize QDIO (layer 2), certain prerequisites must be met. Consult with your system administrator regarding these prerequisites.

Archived

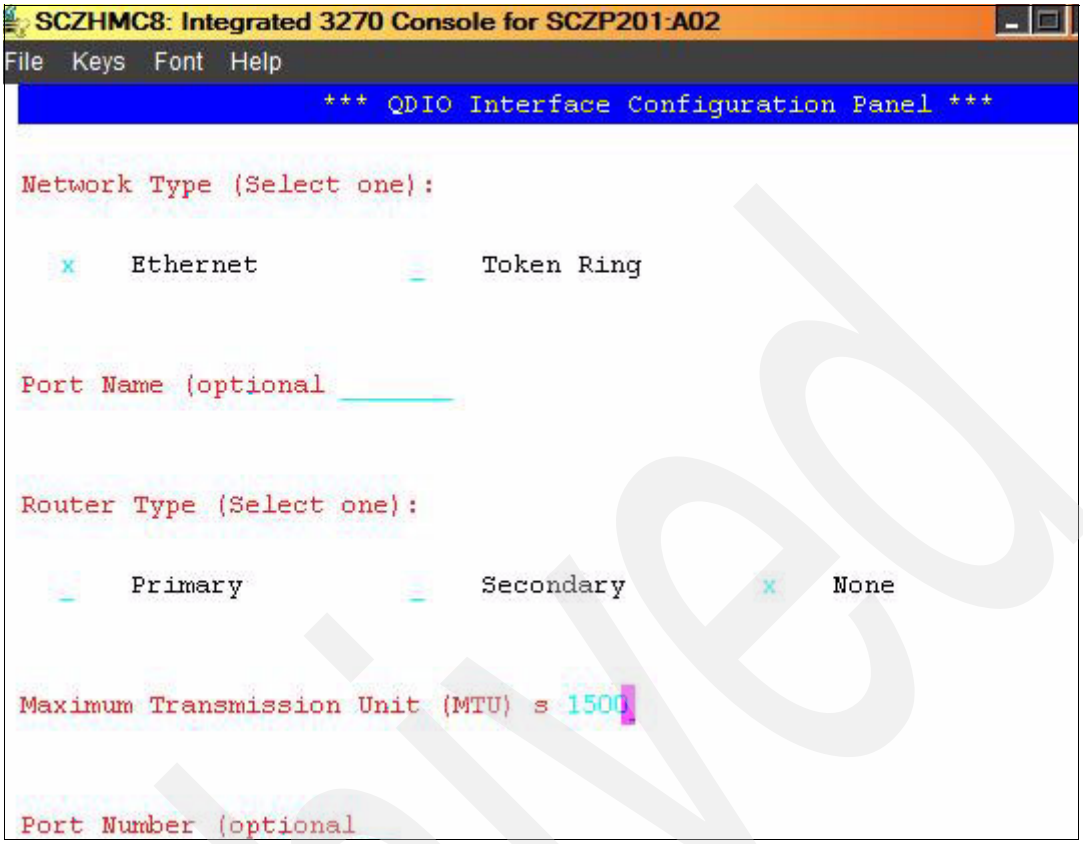


Figure 4-12 IPWIZARD window 3

5. In general, a value for the Port Name is no longer necessary and a Router Type of None is recommended. Press **F5** to complete the wizard.

```
DTCIPW2508I DTCIPWIZ EXEC is attempting to create the necessary
DTCIPW2508I configuration files
```

6. Enter 1 to restart the TCP/IP stack:

```
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 = 2     FORCED BY MAINT
...
Successfully PINGed Interface (9.12.5.22)
Successfully PINGed Gateway (9.12.4.1)
Successfully PINGed DNS (9.12.6.7)
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. At this point your z/VM TCP/IP stack should be running, and you should be able to ping it from another system.

   If the **IPWIZARD** fails, you must continue debugging it until it succeeds. Double-check all values. Verify that the TCP/IP network and OSA information you were given are properly associated.

Archived

HMC Integrated 3270 Console or 3270 emulator? At this point, z/VM should be accessible over the network. You can continue working at the HMC, or you can access your new system through a 3270 emulator.

If you want to switch to 3270 emulator, first logoff from MAINT or issue **DISConnect** on the Integrated 3270 Console. Note the following points:

- If you log off, the session is ended; it is analogous to shutting and powering down a PC.
- If you disconnect, your session remains where it is and is resumed when you log back on; it is analogous to turning off a PC monitor.

In general, use **LOGOFF** from system administration user IDs such as MAINT. However, always **DISCONNECT** from z/VM service machines such as TCPIP and user IDs running Linux, because logging off from them will terminate the service or crash Linux.

4.3 Configuring the XEDIT profile

Logon to MAINT (if not logged on already).

The **XEDIT** command looks for the file XEDIT PROFILE configuration file when it is invoked. Many z/VM user IDs do not have such a personal or shared system file, so all **XEDIT** default values are in effect. The MAINT 191 (A) disk has a PROFILE XEDIT so when you are editing files on MAINT, the values in this profile are usually in effect.

If you have never used XEDIT before, refer to , “XEDIT quick reference sheet” on page 220, for more information about this topic. The z/VM 5.4 PDF library is available on the Web at:

http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm54

Search for the *XEDIT User's Guide and Command Reference*.

One default setting that can be dangerous, especially if you use F12 to retrieve commands, is that PF12 is set to the **FILE** subcommand. Sometimes you may not want to save your changes with the stroke of one key. We recommend that you set PF12 to the **?** subcommand, which has the effect of a retrieve key:

```
==> copy profile xedit a profile xediorig a (oldd
==> x profile xedit a
```

Before:

```
SET PF12 FILE
```

After:

```
SET PF12 ?
```

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

4.4 Customizing the SYSTEM CONFIG file

The first configuration file read when z/VM IPLs is the SYSTEM CONFIG file. The following changes are recommended:

- Change the system name.

- Increase retrieve key capacity.
- Allow virtual disks (VDISKs) to be created.
- Turn off the Disconnect Timeout. This will prevent idle disconnected users from being forced off the system.
- Define a virtual switch (VSWITCH) that will be used for Linux networking.

To make these changes, perform the following steps:

1. To edit the SYSTEM CONFIG file, the MAINT CF1 minidisk must be released as a CP disk through the **CPRELASE** command. The CP disks are queried through the **QUERY CPDISK** command. Note the MAINT CF1 disk is accessed as CP disk A before it is released, but not after.

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  540RES D850 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  540RES D850 CKD         279        398
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF2 MAINT    OCF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  540RES D850 CKD         279        398
```

2. After it is released, you are able to access the MAINT CF1 disk read-write. Use the **LINK** command with multi-read (**MR**) parameter and **ACCESS** command to get read-write access as your F disk.

```
==> link * cf1 cf1 mr
==> acc cf1 f
```

3. Make a backup copy of the vanilla SYSTEM CONFIG file using the **COPYFILE** command with the **OLDDATE** parameter so that the time stamp of the file is not modified. Note that because the target file name (system) and mode (f) are the same, the equal sign (=) can be used as a wildcard.

```
==> copy system config f = conforig = (oldd
```

4. Edit the original file:

```
==> x system config f
```

5. The system name is set to ZVMV5R40 by default in the System_Identifier_Default statement. You can search for it using the / subcommand:

```
====> /System_Identifier_D
```

   Modify this to the new name of your system. In this example, VMLINUXA is used.

```
System_Identifier_Default <VMLINUXA>
```

6. Next look for the Features statement. You can search for it again or you can use **F8** to page down. The following changes and additions are recommended:
   - Increase the number of commands that can be retrieved from 20 to 99.
   - Set the Disconnect_Timeout to **off** so disconnected users do not get forced off.
   - Allow unlimited VDISKs to be created by users by changing Userlim to **infinite** and by adding the **Syslim infinite** clause, as shown here:

```
Features ,
  Disable ,                     /* Disable the following features */
    Set_Privclass ,             /* Disallow SET PRIVCLASS command */
    Auto_Warm_IPL ,             /* Prompt at IPL always           */
    Clear_TDisk   ,             /* Don't clear TDisks at IPL time */
  Retrieve ,                    /* Retrieve options               */
```

```
  Default  99 ,                /* Default.... default is 20      */
  Maximum  255 ,               /* Maximum.... default is 255     */
MaxUsers noLimit ,             /* No limit on number of users    */
Passwords_on_Cmds ,            /* What commands allow passwords? */
  Autolog  yes ,               /* ... AUTOLOG does               */
  Link     yes ,               /* ... LINK does                  */
  Logon    yes ,               /* ... and LOGON does, too        */
Disconnect_Timeout off ,       /* Don't force disconnected users */
Vdisk ,                        /* Allow VDISKS for Linux swaps   */
  Syslim  infinite  ,
  Userlim  infinite
```

7. Define a VSWITCH:

   Use the **BOTTOM** subcommand to go to the bottom of the file. Add some lines (you can use the **XEDIT** add subcommand **a3**). Define a VSWITCH and set the MAC address prefix. This will set the first three bytes of the MAC address created for each virtual NIC. If you have a multiple z/VM systems, increment this value to avoid having identical MAC addresses created. The last three bytes of the MAC address are automatically incremented by z/VM as they are assigned, so they will be unique on each z/VM system. Modify the two starting addresses of the OSA triplets (3024 and 3028 in this example) to those you specified in 2.7.1, “z/VM resources worksheet” on page 16.

```
====> bot
====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev <3024> <3028>
vmlan macprefix 020001
```

8. Save your changes with the XEDIT **FILE** subcommand:

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

9. Test your changes with the **CPSYNTAX** command, which is on the MAINT 193 disk:

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

   Pay attention to the output. If you receive any syntax errors, fix them before proceeding.

10. Release and detach the MAINT CF1 disk with the **RELEASE** command and **DETACH** parameter. Then put it back online with the **CPACCESS** command:

```
==> rel f (det
DASD 0CF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.
```

11. Verify that the CP disk A has been accessed through the **QUERY CPDISK** command:

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  540RES D850 CKD          39        158
MNTCF2 MAINT    0CF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    0CF3  C   R/O  540RES D850 CKD         279        398
```

   Note that all three CP disks are now accessed.

4.5 Configuring TCP/IP to start at IPL time

Configure the TCPIP service machine to be started when z/VM IPLs. This is commonly accomplished from the AUTOLOG1 **PROFILE EXEC**. If the **noautolog** parameter is *not* specified when z/VM is IPLed, then the AUTOLOG1 virtual machine is started.

Because this virtual machine IPLs CMS, the PROFILE EXEC that is found on its A disk is run. This is analogous to the /etc/profile file on Linux and the autoexec.bat on DOS systems.

1. Logoff from MAINT.

```
==> log
```

2. You should see a new logon window. Logon to AUTOLOG1. Again, the password is the same as the user ID.
3. At the VM READ prompt, enter the command **ACCESS (NOPROF** so that the PROFILE EXEC is not run.

```
LOGON AUTOLOG1
z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:06:05 EDT THURSDAY 05/15/08
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V5.4.0    2008-05-13 14:27
==> acc (noprof
```

4. Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.
   a. Use the **VMLINK** command to both link to the disk read-only and to access it as the highest available file mode. The default read password is **read**:

```
==> vmlink maint 191
ENTER READ PASSWORD:
==> read
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

   b. Copy the PROFILE XEDIT to your A disk:

```
==> copy profile xedit z = = a
```

5. Make a backup copy of the PROFILE EXEC and edit it:

```
==> copy profile exec a = execorig =
==> x profile exec
```

6. You should see the text in the top half of the following example. Modify it as follows.
   a. You can safely delete the Address Command line.
   b. Add a line to start the TCPIP user ID through the **XAUTOLOG** command and keep two statements that start the VSWITCH controllers.
   c. Add a line to logoff from AUTOLOG1 when the EXEC is complete. There is no need to keep that virtual machine running, because its sole purpose is to run the PROFILE EXEC.

   Before:

```
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/

Address Command
'CP XAUTOLOG VMSERVS'
```

```
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
```

After:

```
/**************************/
/*  Autolog1 Profile Exec  */
/**************************/
'cp xautolog tcpip'                /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp logoff'                        /* logoff when done */
```

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

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

8. Logoff from AUTOLOG1:

```
==> log
```

When your z/VM system IPLs, the TCP/IP stack should now come up automatically (as long as you do *not* specify the **notautolog** parameter at IPL time).

4.5.1 Renaming the TCPIP configuration file

It is recommended that you change the name of the main TCPIP configuration file from PROFILE TCPIP to <system_ID> TCPIP, where <system_ID> is the name of your new z/VM system. This is to avoid the possibility that the PROFILE TCPIP file will be overwritten when applying maintenance.

1. Logoff from AUTOLOG1 and logon to TCPMAINT. The PROFILE TCPIP file is on the TCPMAINT 198 disk, which is accessed as the D disk.
2. Make a backup copy of the original PROFILE TCPIP, then rename it to <SYSTEM_ID> TCPIP (where <SYSTEM_ID> is VMLINUXA in this example). When the TCP/IP service machine starts, it will search for this file before the file PROFILE TCPIP.

```
==> copy profile tcpip d = tcpiorig = (oldd
==> rename profile tcpip d <vmlinuxa> = =
```

3. You have now backed up and renamed your TCP/IP profile. You can verify through the **LISTFILE** command:

```
==> listfile * * d
VMLINUXA TCPIP    D1
SYSTEM   DTCPARMS D1
PROFILE  TCPIORIG D1
```

4.5.2 Copy the PROFILE XEDIT file

Again copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.

1. Use the **VMLINK** command to both link to the disk read-only and to access it as the highest available file mode. The default read password is **read**:

```
==> vmlink maint 191
```

```
ENTER READ PASSWORD:
read
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

2. Copy the `PROFILE XEDIT` to your A disk:

```
==> copy profile xedit z = = a
```

Now, XEDIT sessions on `TCPMAINT` will have the same configuration as on `MAINT`.

4.5.3 Configuring the FTP server

Turn on the FTP server by editing the renamed configuration file:

1. Add an `AUTOLOG` statement near the top of the file with `FTPSERVE` as the only entry.
2. In the `PORT` statement, remove the semicolons to uncomment the lines with `FTPSERVE` on them (ports 20 and 21). These changes will cause the FTP server to start when TCPIP is started. The important lines before the file is edited and after are shown:

```
==> x <vmlinuxa> tcpip d
```

Before:

```
; ---------------------------------------------------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE DHCPD REXECD SNMPD SNMPQE LDAPSRV
ENDOBEY
; ---------------------------------------------------------------------
PORT
; 20 TCP FTPSERVE  NOAUTOLOG ; FTP Server
; 21 TCP FTPSERVE            ; FTP Server
  23 TCP INTCLIEN            ; TELNET Server
; 25 TCP SMTP                ; SMTP Server
...
```

After:

```
; ---------------------------------------------------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE ROUTED DHCPD REXECD SNMPD SNMPQE
ENDOBEY
; ---------------------------------------------------------------------
AUTOLOG
  FTPSERVE  0
ENDAUTOLOG

PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
; 25   TCP SMTP                ; SMTP Server
...
```

3. Save your changes with the **`FILE`** subcommand:

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

You could continue to configure the system, but at this time we recommend that you test your changes by shutting down and reIPLing the system.

Archived

4.5.4 Shutting down and reIPLing the system

You may want to be able to shut down and reIPL z/VM without having to access the HMC. Often, the HMC will be logged off and thus the Integrated 3270 console (SYSG) will not be available.

Because of these factors, it is useful to use the System Console (SYSC, which has a title of Operating System Messages on the HMC) in order to shut down z/VM and reIPL it without needing to use the console. This console is always accessible, whether you are logged on to the HMC or not.

z/VM messages during both the shutdown and reIPL process will be written to the system console, but often you will be able to ignore them; you just want your system back in a few minutes over the network.

1. Pass the parameter **IPLPARMS CONS=SYSC** to the **SHUTDOWN REPIL** command:

   ```
   ==> shutdown reipl iplparms cons=sysc
   ```

   You will lose your session, but it should come back in a few minutes as described. When your system is back up, perform the following commands:

2. Start a 3270 session and logon as MAINT. This shows that there is TCP/IP access to z/VM.

3. Query the new VSWITCH:

   ```
   ==> q vswitch
   VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
     PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
     VLAN Unaware
     MAC address: 02-00-01-00-00-01
     State: Ready
     IPTimeout: 5        QueueStorage: 8
     RDEV: 3024.P00 VDEV: 3024 Controller: DTCVSW2
     RDEV: 3028.P00 VDEV: 3028 Controller: DTCVSW1  BACKUP
   ```

   You should see that the VSWITCH VSW1 exists, that the OSA devices you specified are being used and that there are two built-in VSWITCH controllers, DTCVSW1 and DTCVSW2.

4. Use the **QUERY VDISK** and **QUERY RETRIEVE** commands to see the changes made to the Features statement in the SYSTEM CONFIG file:

   ```
   ==> q retrieve
   99 buffers available.  Maximum of 255 buffers may be selected.
   ==> q vdisk userlim
   VDISK USER   LIMIT IS INFINITE
   ==> q vdisk syslim
   VDISK SYSTEM LIMIT IS INFINITE,               0 BLK IN USE
   ```

This shows that the changes to the SYSTEM CONFIG file have taken effect.

4.6 Adding paging volumes

The z/VM operating system resides on the first three CP volumes (or one volume, if you are installing onto 3390-9s). z/VM 5.4 is installed with one full paging volume and one full spool volume. A single spool volume is probably adequate for Linux needs, but a single paging volume is probably not. We recommend that you add five paging volumes so you will have a total of six (or one more 3390-9 if installing onto 3390-9s, which will give the same total of 20034 cylinders of page space).

Archived

If you do not have sufficient DASD, this number can be reduced. Having adequate paging space will give you plenty of headroom to add more Linux virtual machines. A rule of thumb for the amount of paging space is to have twice as much as the total of all memory for all running Linux user IDs combined.

4.6.1 Formatting the paging volumes

Before adding paging volumes to the system, the DASD volumes to be used for minidisk space (PERM) and paging space (PAGE) must be formatted. Normally this is done one volume at a time through the **CPFMTXA** command. If you have just a few volumes, that is fine, but when you have many volumes to format, the process of running **CPFMTXA** can become time-consuming and tedious, which can lead to errors.

Therefore, a REXX EXEC named CPFORMAT is provided to allow you to format many volumes with a single command. The source code for this EXEC is located in “The CPFORMAT EXEC” on page 224. It is a wrapper around CPFMTXA. To use this EXEC, each DASD to be formatted must be first attached using the same real device address as the virtual device address (using **ATTACH <realDev> ***).

Note: This EXEC will label the volumes according to the convention described in 2.2.1, “Volume labeling convention” on page 9. 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.

Getting the CPFORMAT EXEC to z/VM

Logoff from MAINT so you will be able to get the MAINT 191 disk in read-write mode through FTP.

Important: At this point, you will need access to the NFS server described in Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85, in order to get the files **CPFORMAT EXEC**.

You must complete the tasks covered in Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85, in order to proceed.

Start an SSH (putty) session to the NFS server and change to the vm/ directory that was created when you untarred the files associated with this book. Verify that the file CPFORMAT.EXEC exists. Note that the directory name will be one of the following, depending on the Linux distribution you are working with:

```
# cd /nfs/virt-cookbook-S10SP2/vm    // if you are working with SLES 10 SP2
# cd /nfs/virt-cookbook-RH52/vm      // if you are working with RHEL 5.2
# ls CPFORMAT*
CPFORMAT.EXEC
```

Now start an FTP session to z/VM. If you get a reply from the FTP server, it shows that you correctly configured it on the z/VM TCPMAINT user ID. Issue the **ASCII** subcommand to be sure the ASCII characters are converted to EBCDIC, and use the **PUT** subcommand to copy the file.

```
# ftp <9.12.5.22>
Name (9.12.5.22:root): maint
331-Password: maint
230-220-FTPSERVE IBM VM Level 540 at VMLINUXA.ITSO.IBM.COM, 15:47:48 EDT TUESDAY
2008-05-13
```

```
230-MAINT logged in; working directory = MAINT 191
...
ftp> ascii
...
ftp> put CPFORMAT.EXEC
...
ftp> quit
```

Using the CPFORMAT EXEC

Log back into MAINT. You should now have access to the CPFORMAT EXEC. You can get brief help on **CPFORMAT** by using a parameter of **?**:

```
==> cpformat ?

Synopsis:

  Format one or a range of DASD as page, perm, spool or temp disk space
  The label written to each DASD is M<t><xxxx> where:
    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)
    <xxxx> is the 4 digit address

Syntax is:
                                              .-PAGE-.
   >>--CPFORMAT--.-rdev--------------.--AS---+-PERM-+--------->><
                 | <---------------< |       '-SPOL-'
                 '-rdev1-rdev2-------'
```

The following example illustrates how to attach five 3390-3 volumes and use **CPFORMAT** to format them as paging space. Refer to the planning work sheets that you filled out in 2.7.2, “z/VM DASD worksheet” on page 17.

For 3390-9 volumes: If you are installing onto 3390-9s, only one more paging volume may be adequate to start. This will give you two full volumes, or the same equivalent of six 3390-3s.

For example, if your z/VM system installed onto DASD at D850-D852 and you plan to put the second paging volume on D853, then only that volume needs to be added.

Rather than using five consecutive DASD addresses, you may consider using DASD from different address ranges in an attempt to locate the paging volumes on different *ranks* in your disk array. This should enable z/VM to page more efficiently.

1. The DASD that will be used for paging volumes are at real addresses **D855, D950, D951, DA50** and **DB50.** First query the DASD devices to see their status:

   ```
   ==> q <d855 d950 d951 da50 db50>
   DASD D855 NWD855  , DASD D950 NWD950  , DASD D951 NWD951  , DASD DA50 NWDA50
   DASD DB50 NWDB50
   ```

2. Attach the devices to MAINT (the last parameter of “*” means the current user ID) through the **ATTACH** command:

   ```
   ==> att <d855 d950 d951 da50 db50> *
   D855 D950 D951 DA50 DB50 ATTACHED TO MAINT
   ```

3. Use the **CPFORMAT** command with the **AS PAGE** parameter:

   ```
   ==> cpformat d855 d950 d951 da50 db50 as page

   Format the following DASD:
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   ```

```
MAINT    D855 MAINT    D855 3390  NWD855 D855          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    D950 MAINT    D950 3390  NWD950 D950          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    D951 MAINT    D951 3390  NWD951 D951          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA50 MAINT    DA50 3390  NWDA50 DA50          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DB50 MAINT    DB50 3390  NWDB50 DB50          0       3339

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PAGE space (y/n)?
y
...
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
D855 D950 D951 DA50 DB50 DETACHED
D855 D950 D951 DA50 DB50 ATTACHED TO MAINT

DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    D855 MAINT    D855 3390  MPD855 D855          0       3339
MAINT    D950 MAINT    D950 3390  MPD950 D950          0       3339
MAINT    D951 MAINT    D951 3390  MPD951 D951          0       3339
MAINT    DA50 MAINT    DA50 3390  MPDA50 DA50          0       3339
MAINT    DB50 MAINT    DB50 3390  MPDB50 DB50          0       3339
```

This formatting job should run for about 10 to 50 minutes, depending on many factors. Meanwhile, you can continue to format more volumes for PERM (minidisk) space, as described in the next section.

4.6.2 Formatting DASD for minidisks

You could wait until **CPFORMAT** of the five paging volumes completes on MAINT, and then format more volumes for PERM or minidisk space. However, you can also get more format jobs going by using a different user ID.

1. Start a new 3270 session and logon as SYSMAINT. Press **Enter** when you get the VM READ prompt:

```
LOGON SYSMAINT
z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:30:28 EDT WEDNESDAY 05/14/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP113S A(191) not attached or invalid device address
DMSACP723I D (192) R/O
```

2. Link to the MAINT 191 disk read-only to pick up the CPFORMAT EXEC. This can be done with the **VMLINK** command (**VMLINK** performs the **LINK** and **ACCESS** commands, with a read-only link and accessing the highest free file mode letter).

```
==> vmlink maint 191
DMSVML2060I MAINT 191 linked as 0192 file mode D
```

3. Query the eight devices that will be used for the controller, LNXMAINT (common CMS files), and for the golden image. In this example they are D856-D857 D952-D957.

```
==> q <d856-d857 d952-d957>
```

```
DASD D856 NWD856  , DASD D857 NWD857  , DASD D952 NWD952  , DASD D953 NWD953
DASD D954 NWD954  , DASD D955 NWD955  , DASD D956 NWD956  , DASD D957 NWD957
```

4. Attach the eight volumes that will be used for the controller, the common CMS disk, and the golden image. In this example, it is the DASD at the following addresses:

```
==> att <d856-d857 d952-d957> *
DASD D856 ATTACHED TO SYSMAINT D856 WITH DEVCTL HYPERPAV BASE
DASD D857 ATTACHED TO SYSMAINT D857 WITH DEVCTL HYPERPAV BASE
DASD D952 ATTACHED TO SYSMAINT D952 WITH DEVCTL HYPERPAV BASE
DASD D953 ATTACHED TO SYSMAINT D953 WITH DEVCTL HYPERPAV BASE
DASD D954 ATTACHED TO SYSMAINT D954 WITH DEVCTL HYPERPAV BASE
DASD D955 ATTACHED TO SYSMAINT D955 WITH DEVCTL HYPERPAV BASE
DASD D956 ATTACHED TO SYSMAINT D956 WITH DEVCTL HYPERPAV BASE
DASD D957 ATTACHED TO SYSMAINT D957 WITH DEVCTL HYPERPAV BASE
```

5. Invoke the **CPFORMAT** command against these volumes using the parameter **as perm**:

```
==> cpformat <d856-d857 d952-d957> as perm

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D856 SYSMAINT D856 3390  NWD856 D856          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D857 SYSMAINT D857 3390  NWD857 D857          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D952 SYSMAINT D952 3390  NWD952 D952          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D953 SYSMAINT D953 3390  NWD953 D953          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D954 SYSMAINT D954 3390  NWD954 D954          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D955 SYSMAINT D955 3390  NWD955 D955          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D956 SYSMAINT D956 3390  NWD956 D956          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D957 SYSMAINT D957 3390  NWD957 D957          0       3339

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PERM space (y/n)?
y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
SYSMAINT D856 SYSMAINT D856 3390  MMD856 D856          0       3339
SYSMAINT D857 SYSMAINT D857 3390  MMD857 D857          0       3339
SYSMAINT D952 SYSMAINT D952 3390  MMD952 D952          0       3339
SYSMAINT D953 SYSMAINT D953 3390  MMD953 D953          0       3339
SYSMAINT D954 SYSMAINT D954 3390  MMD954 D954          0       3339
SYSMAINT D955 SYSMAINT D955 3390  MMD955 D955          0       3339
SYSMAINT D956 SYSMAINT D956 3390  MMD956 D956          0       3339
SYSMAINT D957 SYSMAINT D957 3390  MMD957 D957          0       3339
```

At this point, you should have page volumes being formatted on MAINT and PERM or minidisk volumes being formatted on SYSMAINT.

When completed, you should have eight newly formatted volumes that can be used as minidisks.

Archived

4.6.3 Updating the SYSTEM CONFIG file

Now that the PAGE and PERM volumes are ready for use, they must be added to the SYSTEM CONFIG file so that z/VM can use them. Follow these steps to update the SYSTEM CONFIG file:

1. Logon to MAINT.
2. The following example uses the same steps to access the MAINT CF1 disk read-write that you used earlier:

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  540RES D850 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  540RES D850 CKD         279        398
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
```

   It is useful to remember this sequence of steps.

3. Make a copy of the working SYSTEM CONFIG file using the WRKS suffix convention:

```
==> copy system config f = confwrks =
```

4. Edit the SYSTEM CONFIG file and specify each of the new page volumes (PAGE) by name as CP_Owned. When you system IPLs, it will pick up these as paging volumes.

```
==> x system config f
====> /cp_owned
...
/*****************************************************************/
/*                    CP_Owned Volume Statements                 */
/*****************************************************************/

    CP_Owned   Slot   1  540RES
    CP_Owned   Slot   2  540SPL
    CP_Owned   Slot   3  540PAG
    CP_Owned   Slot   4  540W01
    CP_Owned   Slot   5  540W02
    CP_Owned   Slot   6  <MPD855>
    CP_Owned   Slot   7  <MPD950>
    CP_Owned   Slot   8  <MPD951>
    CP_Owned   Slot   9  <MPDA50>
    CP_Owned   Slot   10 <MPDB50>
    CP_Owned   Slot   11  RESERVED
    CP_Owned   Slot   12  RESERVED
    CP_Owned   Slot   13  RESERVED
...
```

5. Move down to the User_Volume_List section. User volumes (PERM) can be specified individually with the User_Volume_List statement, or with wild cards through the User_Volume_Include statement. If you are using the labelling convention enforced by the **CPFORMAT EXEC**, then add the following single line to include all PERM space as volume labels all begin with MM (the labeling convention used by the CPFORMAT EXEC).

```
====> /user_v
/*****************************************************************/
/*                         User_Volume_List                      */
/* These statements are not active at the present time.  They are*/
/* examples, and can be activated by removing the comment delimeters */
/*****************************************************************/
```

```
User_Volume_Include MM*
/*  User_Volume_List USRPO1    */
/*  User_Volume_List USRPO2    */
...
====> file
```

6. Save your changes with the **FILE** subcommand. Verify the integrity of the changes with the **CPSYNTAX** command:

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

7. After you confirm that there are no syntax errors, put the MAINT CF1 disk back online. The following example shows how you did this previously:

```
==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  520RES 0200 CKD         39         83
MNTCF2 MAINT    OCF2  B   R/O  520RES 0200 CKD         84        128
MNTCF3 MAINT    OCF3  C   R/O  520RES 0200 CKD        129        188
```

4.6.4 Testing the changes

We recommend that you again shut down and reIPL to test the changes. Before you shut down, note that you have only one page volume (540PAG) through the **QUERY ALLOC PAGE** command. Your output should look similar to the following:

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
540PAG D852          1       3338 600840      0      0   0%
                                  ------ ------        ----
SUMMARY                           600840      0          0%
USABLE                            600840      0          0%
```

Now shut the system down again with the command **SHUTDOWN REIPL IPLPARMS CONS=SYSC**. This is analogous to the Linux **reboot** command in that the system attempts to come back up after it shuts down. If you are connected through a 3270 emulator, you will lose your session, but if all goes well, your system will be available again in a few minutes.

```
==> shutdown reipl iplparms cons=sysc
```

After the system comes back, logon as MAINT and look at the page space again. You should see that you now have six paging volumes:

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
540PAG D852          1       3338 600840      0      0   0%
MPD855 D855          0       3338 601020      0      0   0%
MPD950 D950          0       3338 601020      0      0   0%
MPD951 D951          0       3338 601020     12     12   1%
MPDA50 DA50          0       3338 601020      0      0   0%
MPDB50 DB50          0       3338 601020      0      0   0%
```

```
                                    ------ ------        ----
SUMMARY                              3521K     12          1%
USABLE                               3521K     12          1%
```

The output shows there are six paging volumes constituting 3521 KB pages, or about 14 GB of page space (there are 4 KB per page).

4.7 Creating a user ID for common files

Now it is time to define your first z/VM user ID, LNXMAINT. It will be used to store files that will be shared by Linux user IDs. Before starting, make a copy of the original USER DIRECT file:

```
==> copy user direct c = direorig = (oldd
```

4.7.1 Define the user in the USER DIRECT file

A small, 20-cylinder minidisk is allocated at virtual address 191. A larger, 300-cylinder minidisk (approximately 225 MB), which is to be shared by many guests, is defined at virtual address 192. Use the next free DASD designated as PERM space on your worksheet (2.7.2, "z/VM DASD worksheet" on page 17). Cylinder 0 should always be reserved for the label; therefore, you start minidisks at cylinder 1.

1. Edit the USER DIRECT file and add the following user ID definition to the bottom of the file:

```
==> x user direct c
====> bottom
====> a 6
...
USER LNXMAINT LNXMAINT 64M 128M BEG          1
 INCLUDE TCPCMSU                             2
 LINK TCPMAINT 592 592 RR                    3
 MDISK 0191 3390 0001 0020 MMD857 MR READ     WRITE    MULTIPLE   4
 MDISK 0192 3390 0021 0300 MMD857 MR ALL      WRITE    MULTIPLE   5
*                                                                  6
...
====> file
```

   Note the following explanations:

   1 User ID LNXMAINT, same password, default size of 64 MB, with class B, E and G privileges

   2 Include the profile named TCPCMSU (defined earlier in the USER DIRECT file)

   3 Link to the TCPMAINT 592 disk read-only for access to **FTP** and other TCP/IP commands.

   4 Define a 191 minidisk of size 20 cylinders from volume MMD857.

   5 Define 192 minidisk of size 300 cylinders (approximately 225 MB) from volume MMD857 with the special read password of **ALL**, which allows read access from any user ID without a disk password.

   6 An empty comment line for better readability.

2. Whenever an MDISK statement is added or modified in the USER DIRECT file, always check for overlapping cylinders and gaps. (Gaps will only leave empty disk space, but overlaps can occur because unfortunately z/VM will allow you to define multiple minidisks over the *same* disk space). You can perform this check by using the **DISKMAP** command:

```
==> diskmap user
```

```
The minidisks with the END option specified in this directory will not be includ
ed in the following DISKMAP file.

File USER DISKMAP A has been created.
```

3. The file created, `USER DISKMAP`, contains a mapping of all minidisk volumes defined in the `USER DIRECT` file. It will list any overlaps or gaps found on the volumes. Edit the file and turn off the prefix area with the **`XEDIT PREFIX OFF`** subcommand to view 80 columns:

```
==> x user diskmap
====> prefix off
```

4. Search for the text **`overlap`** with the / subcommand:

```
====> /overlap
```

You should see the error message `DMSXDC546E Target not found`. This means that no minidisks are overlapping each other.

Now search for gaps. You should also see some gaps:

```
====> /gap
...
                                          0         500         501    GAP
$$$$$$   DATAMOVE   5F0     3380      00501       00501       00001
         DATAMOVE   5FF     3380      00502       00502       00001

--------------------------------------------------------------------------

VOLUME   USERID     CUU   DEVTYPE     START         END        SIZE
                                          0           0           1    GAP
$$$LNX   ZVMMAPLX   150     3390      00001       03338       03338

--------------------------------------------------------------------------

VOLUME   USERID     CUU   DEVTYPE     START         END        SIZE
                                          0           0           1    GAP
MMD857   LNXMAINT   0191    3390      00001       00020       00020
         LNXMAINT   0192    3390      00021       00320       00300
...
```

Three `GAP`s should be listed on the right side:

- 501 cylinders on the `$$$$$$` volume
- 1 cylinder on the `$$$LNX` volume
- 1 cylinder on volume used for LNXMAINT 191 and 192 disks (`MMD857`, in this example)

There is no need to be concerned about the first two gaps because they are expected, given the layout of the default `USER DIRECT` file. To avoid a 1-cylinder gap being reported on each user volume, we recommend that you use the user ID `$ALLOC$`. This user is set to `NOLOG`, which means it can never be logged onto. Thus, it is not a conventional user ID; instead it is a convenient place to put dummy minidisk definitions for cylinder 0 of all `PERM` volumes.

5. Get out of the file `USER DISKMAP` with the **`QUIT`** command or by pressing **`F3`**.

6. Edit the `USER DIRECT` file again and add a new minidisk definition at virtual address `A04` for the first cylinder of the DASD you added (the label is `MMD857`, in this example):

```
==> x user direct
====> /user $alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 540RES R
 MDISK A02 3390 000 001 540W01 R
 MDISK A03 3390 000 001 540W02 R
 MDISK A04 3390 000 001 <MMD857> R
```

Archived

7. Save your changes and run **DISKMAP** again. Edit the USER DISKMAP file. This time you should see just two gaps for volumes with labels $$$$$$ and $$$LNX. If you search for the $ALLOC$ user ID, you should see the disk map of the volume you added for LNXMAINT:

```
==> diskmap user
==> x user diskmap
====> prefix off
====> /$ALLOC
MMD857   $ALLOC$     A04     3390       00000       00000       00001
         LNXMAINT   0191     3390       00001       00020       00020
         LNXMAINT   0192     3390       00021       00320       00300
...
```

8. When you are done, you can quit without saving changes by pressing **F3**.

```
====> F3
```

9. Now that you are sure the minidisk layout is correct, the changes to the USER DIRECT file can be brought online through the **DIRECTXA** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
```

If the **DIRECTXA** command fails, correct the problem before proceeding.

At this point, you have now defined your first z/VM user ID named LNXMAINT.

4.7.2 Logging and customizing the new user ID

Now you should be able to logon to the new user ID and format its two minidisks.

1. Logoff from MAINT and logon to LNXMAINT.

```
LOGON LNXMAINT
z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 11:32:19 EDT THURSDAY 05/15/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP112S A(191) device error
```

   You should receive an error message ending in device error. You receive this error because when CMS is started, it tries to access the user's 191 minidisk as file mode A. The 191 minidisk has been defined to this user ID; however, it has never been formatted as a CMS file system.

2. To format this disk for CMS, use the **FORMAT** command. It requires a parameter specifying the file mode to access the disk as mode **A**, as shown in the following example:

```
==> format 191 a
DMSFOR603R FORMAT will erase all files on disk A(191). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
lxm191
DMSFOR733I Formatting disk A
DMSFOR732I 20 cylinders formatted on A(191)
```

3. Format the larger 192 disk as the D minidisk, which should take a minute or two:

```
==> format 192 d
DMSFOR603R FORMAT will erase all files on disk D(192). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
lxm192
DMSFOR733I Formatting disk D
DMSFOR732I 300 cylinders formatted on D(192)
```

At this point, you have formatted the two minidisks and accessed them as file modes A and D. You can confirm this through the **QUERY DISK** command:

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        0          7-00       3593       3600
LXM192 192  D   R/W   300 3390 4096        0         11-00      53989      54000
MNT190 190  S   R/O   100 3390 4096      694      14980-83       3020      18000
MNT19E 19E  Y/S R/O   250 3390 4096     1018      28165-63      16835      45000
```

4.7.3 Copying a PROFILE XEDIT

Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.

1. Use the **VMLINK** command to both link to the disk read-only and to access it as the highest available file mode. The default read password is **read**:

```
==> vmlink maint 191
ENTER READ PASSWORD:
==> read
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

2. Copy the PROFILE XEDIT to your A disk:

```
==> copy profile xedit z = = a
```

4.7.4 Creating a PROFILE EXEC

Create a simple PROFILE EXEC that will be run each time this user ID is logged on.

1. Create the new file through XEDIT and add the following lines (be sure to type the **A** file mode so that you do not pick up a PROFILE EXEC on another disk). REXX EXECs must always begin with a C language-style comment.

```
==> x profile exec a
====> a 5
/* PROFILE EXEC */
'acc 592 e'
'cp set run on'
'cp set pf11 retrieve forward'
'cp set pf12 retrieve'
====> file
```

   This PROFILE EXEC accesses the TCPMAINT 592 disk as file mode E, sets CP run on, and sets the retrieve keys per convention.

2. You could test your changes by logging off and logging back on. However, typing the command **PROFILE** will produce the same results.

```
==> profile
DMSACP723I E (592) R/O
```

3. By default, CMS tries to access the 191 disk as A and the 192 disk as D. Also, you should have the TCPMAINT 592 disk accessed as E. Again verify through the **QUERY DISK** command:

```
==> q disk
LXM191 191  A   R/W    20 3390 4096        2          9-01       3591       3600
LXM192 192  D   R/W   300 3390 4096        0         11-00      53989      54000
TCM592 592  E   R/O    70 3390 4096      904      10251-81       2349      12600
MNT190 190  S   R/O   100 3390 4096      694      14980-83       3020      18000
MNT19E 19E  Y/S R/O   250 3390 4096     1018      28165-63      16835      45000
MNT191 120  Z   R/O   175 3390 4096       53        391-01      31109      31500
```

4. Verify that your F11 and F12 keys are set to the **RETRIEVE** command through the **QUERY PFKEYS** command:

```
==> q pf
...
PF10 UNDEFINED
PF11 RETRIEVE FORWARD
PF12 RETRIEVE BACKWARD
...
```

4.7.5 Copying files associated with this book to LNXMAINT

The z/VM files associated with this book are in the vm/ subdirectory of the NFS server you set up earlier. These files should be stored on the larger 192 disk which is accessed as your D disk. Logoff from LNXMAINT so that the 192 disk can be accessed read-write.

Start an SSH session on the NFS server and change directory to the VM files associated with this book. The directory name will be one of the following, depending on the distribution you are working with:

```
# cd /nfs/virt-cookbook-S10S2/vm    // if you are working with SLES 10 SP2
# cd /nfs/virt-cookbook-RH52/vm     // if you are working with RHEL 5.2
```

FTP to z/VM. By default, FTP copies files to your 191 disk, so first change directory to the LNXMAINT 192 disk. The files are in ASCII, so invoke the **ASCII** subcommand to convert them to EBCDIC. Then use the **mput *** subcommand to copy all the files from the vm/ subdirectory to LNXMAINT:

```
# ftp <9.12.5.22>
Connected to 9.12.5.22.
Name (9.12.5.22:root): lnxmaint
331-Password:
Password: lnxmaint
230-LNXMAINT logged in; working directory = LNXMAINT 191
Remote system type is z/VM.
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> prompt
Interactive mode off
ftp> ascii
...
ftp> mput *
...
ftp> quit
```

Logon to LNXMAINT. You should see the following files on your D disk:

```
==> filel * * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=8 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
```

```
CHPW540  XEDIT    D1 V         77        194          3  7/01/08  4:30:39
CPFORMAT EXEC     D1 V         79        252          3  7/01/08  4:30:39
LABEL540 EXEC     D1 V         77        116          2  7/01/08  4:30:39
LABEL540 XEDIT    D1 V         77         50          1  7/01/08  4:30:39
LBL540-9 EXEC     D1 V         77        105          2  7/01/08  4:30:39
LBL540-9 XEDIT    D1 V         77         49          1  7/01/08  4:30:39
PROFILE  EXEC     D1 V         63         17          1  7/01/08  4:30:39
SAMPLE   PARM-S10 D1 V         80          9          1  7/01/08  4:30:39
SLES10S2 EXEC     D1 V         69          9          1  7/01/08  4:30:39
SWAPGEN  EXEC     D1 V         72        358          5  7/01/08  4:30:39
```

4.8 Customizing system startup and shutdown

When your z/VM system is IPLed, it is often desirable to have important Linux systems also start. Conversely, when you shut down z/VM, it is desirable to have all Linux systems shut down first.

4.8.1 Configuring the AUTOLOG1 PROFILE EXEC

We recommend that the following tasks be accomplished by using the `AUTOLOG1 PROFILE EXEC`.

- Configure Linux to shut down gracefully through the **`SET SIGNAL`** command.
- Overcommit memory through the **`SET SRM STORBUF`** command.
- Grant access to the VSWITCH for each Linux user.
- Start user IDs that should be started through the **`XAUTOLOG`** command.
- Limit minidisk cache in main storage and turn it off in expanded storage.

1. Logoff from `LNXMAINT` and logon to `AUTOLOG1`. At the `VM READ` prompt you have usually been pressing Enter, which causes the `PROFILE EXEC` to be run. If you do not want this EXEC to run, enter the command **`ACCESS (NOPROF`**:

```
LOGON AUTOLOG1
z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:06:05 EDT THURSDAY 05/15/08
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V5.4.0    2008-05-13 14:27
==> acc (noprof
```

2. Make a copy of the working `PROFILE EXEC`:

```
==> copy profile exec a = execwrks =
```

3. Edit the file and add the bolded text. A **`LOGOFF`** command is added at the end of the EXEC, so the virtual machine will be logged off when it is complete. This will save a small amount of memory on the system, but does add the requirement that you type **`acc (noprof`** at the VM READ prompt when you log on interactively.

```
==> x profile exec
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'                /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
```

```
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                    /* set the retrieve key */
'cp set mdc stor 0m 128m'            /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'            /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'         /* Allow guests 5 min to shut down */
'cp logoff'                          /* logoff when done */
```

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

Important: The **set mdc** and **set srm** lines are z/VM tuning values. It is believed that these are useful starts for Linux systems, but will not be optimal for all z/VM systems.

For more information about these values, see the following Web sites:

http://www.vm.ibm.com/perf/tips/linuxper.html
http://www.vm.ibm.com/perf/tips/prgmdcar.html

You may choose to modify or omit some of these settings. Your system should now be configured to start up and send a signal to shut down Linux user IDs.

4.8.2 Testing the changes

To test your changes, you must reIPL z/VM again. Perform the following steps:

1. Shut down and reIPL your system.

```
==> shutdown reipl iplparms cons=sysc
SYSTEM SHUTDOWN STARTED
```

2. When your system comes back, logon as MAINT.
3. Query the SRM values to verify that the new STORBUF settings are in effect and the SIGNAL SHUTDOWN value is set to 300 seconds:

```
==> q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=300% Q2=250% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
...
==> q signal shutdown
System default shutdown signal timeout: 300 seconds
```

This output shows that your changes have taken effect.

4.9 Addressing z/VM security issues

This section briefly discusses the following security issues:

- z/VM security products
- High level z/VM security
- Linux user ID privilege classes
- z/VM user ID and minidisk passwords

4.9.1 VM security products

You might want to use a z/VM security product such as IBM RACF® or CA VM:Secure. They allow you to address more security issues such as password aging and the auditing of user access attempts.

4.9.2 High level z/VM security

The paper *z/VM Security and Integrity* discusses the isolation and integrity of virtual servers under z/VM. It is available on the Web at:

http://www-1.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130145.pdf

Linux user ID privilege classes

Another security issue is the privilege class that Linux user IDs are assigned. The IBM Redpaper publication *Running Linux Guests with less than CP Class G Privilege* addresses this issue. It is on the Web at:

http://www.redbooks.ibm.com/redpapers/pdfs/redp3870.pdf

z/VM user ID and minidisk passwords

All passwords in a "vanilla" z/VM system are the same as the user ID. This is a large security hole. The *minimum* you should do is to address this issue.

There are two types of passwords in the `USER DIRECT` file:

User IDs The password required to logon with

Minidisks Separate passwords for read access, write access and multi-write access

Both types of passwords should be modified. This can be done using the `CHPW540 XEDIT` macro defined in the next section.

4.9.3 Changing passwords in USER DIRECT

Changing the passwords can be done manually in **`XEDIT`**. However, this is both tedious and error-prone. So a profile named `CHPW540 XEDIT` has been included with this book. The source code is in , "The CHPW540 XEDIT macro" on page 228.

This macro will change all z/VM passwords to the same value, which may still not be adequate security, given the different function of the various user IDs. If you want different passwords, you have to modify the `USER DIRECT` file manually, either with or without using the `CHPW52 XEDIT` macro.

To modify all user ID and minidisk passwords to the same value, perform the following steps.

1. Logon to `MAINT`.
2. Link and access the `LNXMAINT 192` disk to pick up the **`CHPW540 XEDIT`** macro:

```
==> vmlink lnxmaint 192
DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z
```

3. Make a backup copy of the `USER DIRECT` file and first be sure the password that you want to use is not a string in the file. For example, if you want to change all passwords to `lnx4vm`, then do the following:

```
==> copy user direct c = direwrks = (oldd
==> x user direct c
====> /lnx4vm
```

```
DMSXDC546E Target not found
====> quit
```

The `Target not found` message shows that the string `lnx4vm` is not used in the `USER DIRECT` file, so it is a useful candidate for a password.

4. Edit the `USER DIRECT` file with a parameter of **(profile chpw540)** followed by the new password. Rather than invoking the default profile of `PROFILE XEDIT`, this command will invoke the **XEDIT** macro named `CHPW540 XEDIT` and pass it the new password. For example, to change all passwords to `lnx4vm`, enter the following command:

```
==> x user direct c (profile chpw540) lnx4vm

Changing all passwords to: LNX4VM

DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
...
```

5. When the profile finishes you are left in the **XEDIT** session with all passwords modified. You may wish to first examine the changes. Then save the changes with the **FILE** subcommand:

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

6. Bring the changes online with the **DIRECTXA** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
```

Your new directory is online. Remember the new password.

Note that this **XEDIT** macro will only work on a vanilla `USER DIRECT` file because it searches for the original user IDs next to passwords. If you want to change your password again, it should be much easier because you can use the **XEDIT CHANGE** subcommand. For example, to change all passwords from `lnx4vm` to `vm53nx`, invoke the following commands:

```
==> x user direct c
====> c/LNX4VM/VM4LNX/* *
DMSXCG517I 798 occurrence(s) changed on 345 line(s)
```

At this point, your z/VM system is now customized and ready for Linux. We recommend that you back up your system to tape.

4.10 Backing up your z/VM system to tape

Your system is now customized with a running TCP/IP stack, a highly available VSWITCH, a startup and shutdown process, and a user ID for shared files. You have changed the passwords. This would be an appropriate time to back up the system to tape.

There are five system volumes that should be backed up: `540RES`, `540SPL`, `540PAG`, `540W01` and `540W02` (or just the first three, if you are using 3390-9s). You also have configured a sixth volume that is important to Linux; that is, the first 320 cylinders of the volume with `LNXMAINT` on it.

Archived

To learn how to back up these volumes to tape, refer to Step 11, Store a backup copy of the z/VM system on tape, which you can find in Chapter 8 “Load the system image” in *z/VM Guide for Automated Installation and Service*, GC204-6099.

4.11 Relabeling the system volumes

Relabeling the system volumes is optional, but recommended. There are times when you will want to change the volume labels of the five z/VM system volumes (or three, if you installed onto 3390-3s). If there is a possibility that another vanilla z/VM system with the same labels is installed onto volumes accessible by your z/VM system, then one of the systems will not IPL correctly.

When installing z/VM, it is possible to modify all but one volume label, that of the `540RES` volume. This alleviates the problem that is described next, but it does not alleviate the problem of duplicate volume names.

To understand this possibility, refer to Figure 4-13. The z/VM system with the lower device addresses starting at `E340` should IPL without any problem (although you may see a warning at system startup time about duplicate volume labels). However, if the z/VM system starting at device address `F000` is IPLed, the `540RES` volume will be used, but the remaining volumes in the system are searched for by volume label, not by device address. Because the z/VM system 1 addresses are lower than the z/VM system 2 addresses, system 2 will be using the system 1 volumes, which is undesirable for either system.

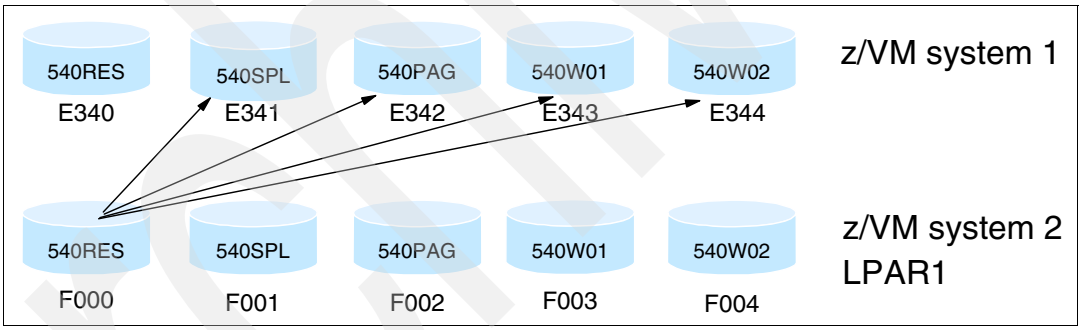


Figure 4-13 The problem with two z/VM systems with identical volume labels

If there is a possibility of another z/VM system being installed on DASD that this system will have access to, it is recommended that you perform the following steps. You will need access to the HMC to perform them:

- Modifying labels in the SYSTEM CONFIG file
- Modifying labels in the USER DIRECT file
- Changing the labels on the five volumes
- Shutting down your system and restarting it

Important: This process *must* be performed as documented, because making a mistake in one of the steps can easily produce an unusable system. Proceed carefully and your system will come back with no problems. Try to do all steps in succession in a short amount of time.

4.11.1 Modifying labels in the SYSTEM CONFIG file

An HMC 3270 session is needed because z/VM will have to be restarted with a FORCE option.

1. Start an Integrated 3270 Console session on the HMC from the CPC Recovery (or just Recovery) menu.
2. If you have not already done so, logon to MAINT and link and access the LNXMAINT 192 disk to pick up the **LABEL540 EXEC** and **XEDIT** macro:

```
==> vmlink lnxmaint 192
DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z
```

3. Note the first five CP-owned volumes through the **QUERY CPOWNED** command:

```
==> q cpowned
   1  540RES  D850  Own    Online and attached
   2  540SPL  D851  Own    Online and attached
   3  540PAG  D852  Own    Online and attached
   4  540W01  D853  Own    Online and attached
   5  540W02  D854  Own    Online and attached
   6  MPD855  D855  Own    Online and attached
   7  MPD950  D950  Own    Online and attached
   8  MPD951  D951  Own    Online and attached
   9  MPDA50  DA50  Own    Online and attached
  10  MPDB50  DB50  Own    Online and attached
  11  ------  ----  -----  Reserved
  12  ------  ----  -----  Reserved
  ...
```

For 3390-9s: If z/VM is installed onto 3390-9s, there should only be three system volumes:

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  540RES  9300  Own    Online and attached
   2  540SPL  9301  Own    Online and attached
   3  540PAG  9302  Own    Online and attached
   ...
```

4. The labeling convention described in 2.2.1, “Volume labeling convention” on page 9 suggests using M in the second character of the label. An **XEDIT** macro, LABEL540 XEDIT, is supplied to help make this process more reliable. It can be used on both the SYSTEM CONFIG and USER DIRECT files.

   To modify the labels in the SYSTEM CONFIG file, release the A CP-disk and access it read-write. Back up the SYSTEM CONFIG file:

```
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
==> copy system config f = confwrks = (oldd rep
```

5. Edit the SYSTEM CONFIG file with the **LABEL540 XEDIT** macro passing the five addresses of the z/VM system volumes (d850-d854, in this example):

```
==> x system config f (profile label540) <d850 d851 d852 d853 d854>
DMSXCG517I 3 occurrence(s) changed on 3 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

```
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

For 3390-9s: If z/VM is installed onto 3390-9s, there is a macro named **LBL540-9 XEDIT** that takes only three parameters. For example, if the system volumes are 9300-9302, the command to edit the SYSTEM CONFIG file would be as follows:

```
==> x system config f (profile lbl540-9) <9300 9301 9302>
DMSXCG517I 3 occurrence(s) changed on 3 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

6. Clear the window and you will be left in XEDIT, editing the file. Search for the string cp_owned and you should see the new labels. Be sure they are correct before saving the file with the **FILE** subcommand:

```
====> /cp_owned
/*                    CP_Owned Volume Statements                    */
/*******************************************************************/

 CP_Owned   Slot   1  MVD850
 CP_Owned   Slot   2  MVD851
 CP_Owned   Slot   3  MVD852
 CP_Owned   Slot   4  MVD853
 CP_Owned   Slot   5  MVD854
 CP_Owned   Slot   6  MPD855
 CP_Owned   Slot   7  MPD950
 CP_Owned   Slot   8  MPD951
 CP_Owned   Slot   9  MPDA50
 CP_Owned   Slot   10 MPDB50
 CP_Owned   Slot   11  RESERVED
 CP_Owned   Slot   12  RESERVED
    ...
====> file
```

7. Verify that there are no syntax errors:

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

8. Release and detach the F disk, **CPACCESS** the A disk, and verify:

```
==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
Ready; T=0.01/0.01 09:19:57
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  540RES D850 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  540RES D850 CKD         279        398
```

You have now changed the labels of the system volumes in the SYSTEM CONFIG file.

Important: It is critical to continue this process now, because your system is currently in a state where it will not IPL cleanly.

Archived

4.11.2 Modifying labels in the USER DIRECT file

In this section you will modify the system volume labels in the USER DIRECT file.

1. Modify the USER DIRECT file again using the **LABEL540 XEDIT** macro. You should see many more occurrences of the labels being changed:

```
==> copy user direct c = direwrks = (oldd rep
==> x user direct c (profile label540) <d850 d851 d852 d853 d854>
DMSXCG517I 99 occurrence(s) changed on 99 line(s)
DMSXCG517I 68 occurrence(s) changed on 68 line(s)
DMSXCG517I 124 occurrence(s) changed on 124 line(s)
DMSXCG517I 2 occurrence(s) changed on 2 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

For 3390-9s: Again, the macro named **LBL540-9 XEDIT** should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to edit the SYSTEM CONFIG file would be as follows:

```
==> x user direct c (profile lbl520-9) <9300 9301 9302>
DMSXCG517I 287 occurrence(s) changed on 287 line(s)
DMSXCG517I 2 occurrence(s) changed on 2 line(s)

DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

You may choose to traverse the file before saving the changes with the **FILE** subcommand:

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

You have now changed the labels of the system volumes in the USER DIRECT and SYSTEM CONFIG files. Again, it is critical that you proceed with the remaining steps.

4.11.3 Changing the labels on the five volumes

In this section you will change the labels on the five volumes through the **CPFMTXA** command. You could do this one volume at a time with the **CPFMTXA LABEL** command. However, the **LABEL540** EXEC has been written to make this step easier.

1. Use the **LABEL540 EXEC** with the physical addresses of the five system volumes:

```
==> label540 <d850 d851 d852 d853 d854>
The volumes are:
DASD D850 CP OWNED  540RES   59
DASD D851 CP OWNED  540SPL   1
DASD D852 CP OWNED  540PAG   0
DASD D853 CP OWNED  540W01   61
DASD D854 CP OWNED  540W02   25

The system volume labels will become:
MVD850 MVD851 MVD852 MVD853 MVD854

ARE YOU SURE you want to relabel the DASD (y/n)?
y
HCPCCF6209I INVOKING ICKDSF.
...
```

For 3390-9s: The **LBL540-9 EXEC** should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to relabel the system volumes would be as follows:

```
==> lbl540-9 <9300 9301 9302>
The volumes are:
DASD 9300 CP OWNED  540RES   142
DASD 9301 CP OWNED  540SPL   2
DASD 9302 CP OWNED  540PAG   0

The system volume labels will become:
MV9300 MV9301 MV9302

ARE YOU SURE you want to relabel the DASD (y/n)?
y
...
```

2. Now that the five volumes have been relabeled (this is sometimes called “clipping the volumes”, which is derived from a contraction (clp) of the z/OS term change label program), you can run the **DIRECTXA** command to update the directory:

   ```
   ==> directxa user
   z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
   EOJ DIRECTORY UPDATED
   HCPDIR494I User directory occupies 43 disk pages
   Ready(00005); T=0.01/0.01 15:45:51
   ```

   A return code of 5 is expected because the labels in the USER DIRECT file are different from the spool data in the currently running system.

Finally, you are ready to issue a **SHUTDOWN** command.

4.11.4 Shutting down your system and restarting it

You will need an HMC console session for this step, if you are not already running from there. To test the changes, you must shut your system down and then restart it. You cannot do a SHUTDOWN REIPL in this situation because you will have to do a **FORCE** start

```
==> shutdown
SYSTEM SHUTDOWN STARTED
HCPSHU960I System shutdown may be delayed for up to 210 seconds
```

Perform the following steps to bring the system back up:

1. Open an HMC session.
2. Select your LPAR.
3. Use the circular arrow racetrack buttons to get to the CPC Recovery (or just Recovery) menu.
4. Double-click the **Integrated 3270 Console** menu item. A new window should appear.
5. Double-click the **LOAD** menu item. The Load Address (D850, in this example) and Load Parameter (SYSG) fields should be correct from the previous IPL.
6. Select the **Clear** radio button. The Load Address and Load Parameter fields should be correct from the previous IPL. Click **OK**.
7. Click **Yes** on the Load Task Confirmation window.

8. Go back to the Integrated 3270 console. After a few minutes, the Standalone Program Loader window should appear. Use the **Tab** key to traverse to the section IPL Parameters and enter the value **cons=sysg**.
9. Press the **F10** key to continue the IPL of your z/VM system. This should take 1 to 3 minutes.
10. At the Start prompt you have to specify a **FORCE** start, again because the spool volume label has changed:

```
==> force drain
```

11. Do not change the time-of-day clock:

```
==> no
```

12. When the IPL completes, **DISCONNECT** from the OPERATOR user ID:

```
==> disc
```

13. Close the HMC windows.
14. Start a 3270 emulator session as the TCPIP service machine should be up. Logon as MAINT.

    Get a 3270 session as MAINT and verify the volume labels have changed with the **QUERY CPOWNED** command:

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  MVD850  D850  Own    Online and attached
   2  MVD851  D851  Own    Online and attached
   3  MVD852  D852  Own    Online and attached
   4  MVD853  D853  Own    Online and attached
   5  MVD854  D854  Own    Online and attached
   6  MPD855  D855  Own    Online and attached
  ...
```

Important: In the event that you IPLed a system with duplicate system volumes, it is possible that you may have destroyed your saved segments. You will know this is the case when you cannot **IPL CMS**. Rather, you will have to **IPL 190**.

To rebuild saved segments, use the following commands (only do so if your saved segments have been destroyed):

```
==> vmfsetup zvm cms
==> sampnss cms
==> ipl 190 clear parm nosprof instseg no
==> acc (noprof
==> acc 5e6 b
==> acc 51d d
==> vmfbld ppf segbld esasegs segblist ( all
```

4.12 Restoring your z/VM system from tape

It is a good idea to *practice* restoring a system before you need to do it under pressure in a real production environment.

Restoring a z/VM system from tape that has the same set of volume labels as the system that is running is problematic. If there are two z/VM systems on the same LPAR with the same volume labels, both systems cannot be IPLed cleanly.

Perform this step only if you successfully completed 4.10, “Backing up your z/VM system to tape” on page 62, and 4.11, “Relabeling the system volumes” on page 63. If you have done both, then the system on tape has volume labels of `540xxx` and the system on DASD has volume labels `MVyyyy`.

You can restore this system to five other 3390-3s, as explained in Appendix E, “Restore the z/VM system backup copy from tape” in *z/VM Guide for Automated Installation and Service*, GC204-6099.

Archived

Archived

5

Servicing z/VM

This chapter describes how to apply:

- A Programming Temporary Fix (PTF)
- A Recommended Service Upgrade (RSU) from “envelope files”

Both processes are basically the same.

Note that this chapter is based on z/VM 5.2, and not on 5.4, due to the timing of the writing of this book. However, the process has not changed significantly, so it should be just as useful. Keep in mind, however, that your output messages and so on will be slightly different.

Important: When applying service, there is always a chance that you may want to back it out. We recommend that you have an up-to-date backup of your system available before starting this section.

The application of corrective service to z/VM is covered in *z/VM V5R3 Service Guide* and in *VMSES/E Introduction and Reference*. Both of these documents can be downloaded in PDF format from the following URL:

`http://www.vm.ibm.com/library`

VMSES/E is a component of z/VM that provides the **SERVICE** and **PUT2PROD** EXECs. The **SERVICE** EXEC performs the following tasks:

- It installs an RSU or applies CORrective service for z/VM components, features, or products.
- It displays either the RSU level of the component specified or whether a particular PTF or APAR has been applied (when used with STATUS).
- It 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 DDR into production.

© Copyright IBM Corp. 2008. All rights reserved.

5.1 Applying a PTF

You may determine that you need to apply a specific fix or PTF to your system.

For example, an Authorized Program Analysis Report (APAR), VM63895, was opened to address the problems reported with virtual NIC support. There are three known symptoms addressed by this APAR:

- Linux guests may lose connectivity after **shutdown -r now** (or any device reset).
- Using an External Security Manager (ESM) to authorize a VLAN list may lead to an FRF002 abend.
- Virtual Hipersockets NIC configured with “VLAN nnn” (exploiting Set Global VLAN ID) did not really filter inbound frames.

The APAR was assigned the following PTF numbers:

- z/VM 5.1.0 VM63895 UM31612
- z/VM 5.2.0 VM63895 UM31613

There are more details on:

http://www-1.ibm.com/support/docview.wss?uid=isg1VM63895

So, for z/VM 5.2, apply PTF UM31613 as described in the next section.

5.1.1 Determining whether a PTF has been applied

Check to make sure the PTF has not previously been applied.

1. Logon to MAINT and issue the **VMFSETUP** command to set up minidisks for TCP/IP and link to them:

```
==> vmfsetup zvm cp (link
VMFSET2760I VMFSETUP processing started for ZVM CP
VMFUTL2205I Minidisk|Directory Assignments:
            String    Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALMOD  E      R/W  2C4   MNT2C4
VMFUTL2205I LOCALSAM  F      R/W  2C2   MNT2C2
...
VMFSET2760I VMFSETUP processing completed successfully
```

2. Use the **VMFINFO** command to query the Software Inventory files. Move the Tab key to ZVM and type s to select it on the PPF Fileid window:

```
==> vmfinfo

PPF Fileid - Help

  Product parameter files (PPFs) define the environment and key variables
  required to process the queries.  The following is a list of all PPFs
  found on all accessed disks.  Select one to continue.  The View function
  can be used to examine one or more PPFs.

  Type a "V" next to one or more PPFs to view their contents, or type an
  "S" next to one PPF to select.

       Options: S - select  V - view

  Option   PPF Fileid
    _      $5654260 PPF      D1
```

```
_       SEGBLD   PPF      D2
_       SERVP2P  PPF      D1
_       UCENG    PPF      D2
s       ZVM      PPF      D2
_       4OSASF40 PPF      D1
```

3. Because the description of the PTF cites a component name of “VM CP”, select **CP** on the Component Name window.
4. Select **PTFs/APARs** on the VMFINFO Main window.
5. Type in the PTF number `UM31613` in the PTF number field, then select **Status of PTF** on the PTF/APAR Queries window:

```
PTF/APAR Queries

 Enter a PTF or APAR number and type an option code.  Then press Enter.
  PPF fileid ...... ZVM       PPF  D
  Component name .. CP                Setup ... NO
  Product ID .....: 5VMCPR20          System .. VM
  PTF number ...... UM31613
  APAR number .....

     Options: S - select
  Option   Query
    s      Status of PTF
    _      Requisites/supersedes of PTF
    _      Dependencies/superseding of PTF
    _      User memo of PTF
    _      Serviceable parts included by PTF

    _      Abstract of APAR(s)
```

6. If the PTF has not been successfully applied, you should see the message `No data found`:

```
Query Output - PTF Status

PPF fileid .....: ZVM       PPF  D
  Component name .: CP                Setup ..: NO
  Product ID .....: 5VMCPR10          System .: VM
 ------------------------------------------------
WN:VMFSIP2481W No entries match search arguments
WN:            TDATA :PTF UM31198
WN:            in table 5VMCPR10 SRVRECS J

No data found
```

This shows that PTF UM31613 has *not* been successfully applied. The sections that follow describe how to obtain and apply it.

5.1.2 Getting service through Internet FTP

You may obtain service for z/VM through tapes. However, you may also want to obtain service over the Internet. If so, point a Web browser to:

`https://techsupport.services.ibm.com/server/login`

If you have an IBM user ID and password, use that. If you do not, you can fill out the form to create an IBM ID and password. You should then be at the following Web site:

`https://www.ibm.com/account/profile/us`

Archived

1. Click **Support and Downloads** at the top menu.
2. Click **Downloads and Drivers** on the left frame.
3. Under Category, select **zSeries (mainframe)**
4. Under Operating Systems, select **z/VM** and click **Software Only**. This should take you to a page entitled Support for VM.
5. Click on **Download selective fixes by PTF**. You may be prompted for your IBM ID and password.
6. In the text box Enter PTF numbers below [e.g: U412345, U467890], enter `UM31613`. All other defaults should be correct. Click **Continue**.
7. In the Verify Order page, click **Submit**. You should get a message similar to the following

```
Your order has been submitted for processing. Email will be sent to nospam@us.ibm.com.

COER NUMBER is <390473266>. This number is used to submit your request. You will receive
a confirmation email that contains your ORDER NUMBER.
...
```

5.1.3 Downloading the service to z/VM

You should receive two e-mails. The first e-mail contains your order number. The second e-mail contains instructions about how to download the service files. Make sure you have access to these. Following is an example.

```
TEXT    = Data sent via "INET". To retrieve your service:
TEXT    =   FTP to: ptf.boulder.ibm.com
TEXT    =   Log on using userid "owte8a" and password "h2q9nep9"
TEXT    =   Enter the following FTP commands:
TEXT    =     cd /390268476/c568411202
TEXT    =     ascii
TEXT    =     get ftp8476.txt
TEXT    =     binary f 1024
TEXT    =     get rlst1585.bin
TEXT    =     get rptf1585.bin
```

1. Logon to MAINT.
2. The `MAINT 500` disk should have a large amount of free space, so it is a useful minidisk on which to download the files. By default, the FTP client saves files on the A disk, so access the 500 disk as A:

```
==> vmlink tcpmaint 592
DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
==> acc 500 a
DMSACC724I 500 replaces A (191)
```

3. Use the FTP client to get the PTF *envelope files* from the Internet. The envelope files can be large, so this may take some time. As you are downloading the files, note the file sizes. Following is an example:

```
==> ftp ptf.boulder.ibm.com
ftp> <owte8a>
ftp> <h2q9nep9>
ftp> cd </390268476/c568411202>
ftp> ascii
ftp> get ftp1585.txt
...
ftp> binary f 1024
ftp> get vlst1585.bin
...
```

```
150 Opening BINARY mode data connection for vlst1585.bin (7168 bytes).
7168 bytes transferred in 0.231 seconds. Transfer rate 31.03 Kbytes/sec.
ftp> get vptf1585.bin
...
551936 bytes transferred in 22.272 seconds. Transfer rate 24.78 Kbytes/sec.
ftp> quit
```

4. Use the **BROWSE** command to view the first text file and verify that the correct number of bytes were downloaded for each file. Press the **F3** key to quit.

```
==> browse ftp1585 txt
VM PTF Package Information
--------------------------
This file contains byte counts of files to receive and instructions
for preparing the files for installation. The byte counts listed
below should match the byte counts of the files when they are received
using FTP.

FILE BYTE COUNTS
----------------

The vptf1585.bin byte count is: 551936.
The vlst1585.bin byte count is: 7168.

Match these byte counts to that reported during the FTP get.
...
====> F3
```

5. You should now have the service or envelope files on your z/VM system. Rename the file type from BIN to SERVLINK, because this is the file type that the **SERVICE** command expects.

```
==> rename vlst1585 bin a = servlink =
==> rename vptf1585 bin a = servlink =
```

6. The envelope files arrive in a compressed format to speed downloads. To use them, they must first be uncompressed with the **DETERSE** command. Use the **(REPLACE** parameter to uncompress them in place and save disk space:

```
==> deterse vlst1585 servlink a = = = (replace
==> deterse vptf1585 servlink a = = = (replace
```

5.1.4 Receiving, applying, and building service

You must receive, apply, and build the PTF. Then it can be put into production. This can be achieved using a process that is much easier now with the **SERVICE** command.

To prepare to use the **SERVICE** command, you must have a 256 MB virtual machine and you must have the minidisk with a large amount of free space; that is what the MAINT 500 minidisk is for.

1. Increase the size of the MAINT virtual machine with the **DEFINE STORAGE** command:

```
==> def stor 256M
STORAGE = 256M
Storage cleared - system reset.
```

2. ReIPL CMS:

```
==> ipl cms
IPL CMS
z/VM V5.4.0    2008-05-13 14:27
==> Press Enter
```

Archived

3. The **SERVICE** command will write to the current A disk. Again access minidisk 500 as A:

```
==> acc 500 a
DMSACC724I 500 replaces A (191)
```

4. Use the **SERVICE ALL** command, specifying the envelope files you downloaded. Many, many screens of output will scroll by and the screens will automatically be cleared. Important messages will be saved to the A (500) disk. This process may take many minutes. Following is an example:

```
==> service all vptf1585
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

A return code of zero (0) is ideal. In general a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

5. The output files written to the A disk are of the form $VMF* $MSGNUM. You may wish to inspect these files.

```
==> filel $VMF* $MSGLOG
MAINT    FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      $VMFSRV  $MSGLOG  A1 V         80        132          3  1/31/06 12:58:09
      $VMFBLD  $MSGLOG  A1 V         80         76          2  1/31/06 12:57:34
      $VMFAPP  $MSGLOG  A1 V         80         70          1  1/31/06 12:57:13
      $VMFREC  $MSGLOG  A1 V         80         55          1  1/31/06 12:57:12
      $VMFMRD  $MSGLOG  A1 V         80         30          1  1/31/06 12:57:10
```

6. Invoke the **VMFVIEW SERVICE** command to review the results of the previous **SERVICE** command. Press the **F3** key to quit. Following is an example:

```
==> vmfview service
********************************************************************************
****            SERVICE              USERID: MAINT                 ****
********************************************************************************
****          Date: 11/31/07 Time: 12:57:09           ****
********************************************************************************
====> F3
```

Ideally there will be no output (as in this example); that means the service applied perfectly.

5.1.5 Putting the service into production

Use the **PUT2PROD** command to put the service into production.

> **Important:** If you run **PUT2PROD** from a 3270 emulator session, you may lose your connection because the TCP/IP service machine may be recycled. Therefore, you may want to run this command from a console.
>
> In this example, applying PTF UM31613 did *not* affect the emulator session.

```
==> put2prod
RDR FILE 0016 SENT FROM MAINT    CON WAS 0016 RECS 0004 CPY  001 T NOHOLD NOKEEP
VMFP2P2760I PUT2PROD processing started
VMFP2P2760I PUT2PROD processing started for VMSES
VMFSET2760I VMFSETUP processing started for SERVP2P VMSESP2P
...
```

```
USER DSC   LOGOFF AS  BLDCMS   USERS = 7     FORCED BY MAINT
VMFP2P2760I PUT2PROD processing completed successfully for SAVECMS
VMFP2P2760I PUT2PROD processing completed successfully
```

Your PTF should now be “put into production”. You may or may not have to reIPL the system, depending on the nature of the PTF applied. It is safest to reIPL through the **SHUTDOWN REIPL** command to completely test the changes:

```
==> shutdown reipl iplparms cons=sysc
SYSTEM SHUTDOWN STARTED
...
```

Your z/VM system should come back in a few minutes. You may want to perform the steps described in 5.1.1, “Determining whether a PTF has been applied” on page 72, to verify that this PTF has been applied.

5.2 Applying a Recommended Service Upgrade

Applying a Recommended Service Upgrade (RSU) is very similar to applying a PTF, as described in 5.1, “Applying a PTF” on page 72. z/VM service can be preventive (RSU) or corrective (COR).

Part 4, “Service Procedure”, in *Guide for Automated Installation and Service,* provides a complete description of how to apply service to z/VM; however, it assumes you are starting with the RSU tape. Following is an example of upgrading to a z/VM 5.3 RSU with the medium being files, not tape.

The following sections summarize how to apply service and also describe how to obtain service through envelope files over the Internet.

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

```
==> q cplevel
z/VM Version 5 Release 3.0, service level 0701 (64-bit)
Generated at 11/18/07 23:44:09 EST
IPL at 11/19/07 13:44:37 EST
```

The service level four digits are split in half, where the first half is the last two digits of the year and the second half is the service level for that year. Therefore, a service level of 0701 means the first service level for the year 2007 has been applied. If you installed from DVD, that was on the second RSU DVD disc.

The overall steps in applying a service level are as follows:

- Getting service from the Internet
- Download the service to z/VM
- Receive, apply, and build the service
- Put the service into production

5.2.1 Getting service from the Internet

When getting service from the Internet, note the following information:

- The PTF number for the most current RSU for z/VM 5.4.0 is UM97540.
- The PTF number for the most current RSU for z/VM 5.3.0 is UM97530.
- The PTF number for the most current RSU for z/VM 5.2.0 is UM97520.

Point a Web browser to:

https://techsupport.services.ibm.com/server/login

1. If you have an IBM user ID and password, use that. If you do not, you can fill out the form to create an IBM ID and password. You should then be at the following Web site:

   https://www.ibm.com/account/profile/us

2. Click **Support and Downloads** at the top menu.
3. In the Choose support type field, choose **System z**.
4. On the Support for mainframes page, choose **z/VM** in the Hardware or operating system field.
5. Click the **Download** tab, then click **Download specific fixes** (these are ordered by PTF number). You may be prompted for your IBM ID and password.
6. In the text box Enter PTF numbers below [e.g: U412345, U467890], enter `UM97530` for the latest z/VM 5.3 service level, or enter `UM97520` for the latest z/VM 5.2 service level, or enter the appropriate PTF number. All other defaults should be correct.
7. Click **Continue**.
8. In the Verify Order page, click **Submit**. You should receive a message similar to the following:

```
Your order has been submitted for processing. Email will be sent to
nospam@us.ibm.com.

COER NUMBER is <390777166>. This number is used to submit your request. You will
receive a confirmation email that contains your ORDER NUMBER.
```

5.2.2 Downloading the service to z/VM

You should receive two e-mails. The first e-mail contains your order number. The second e-mail contains instructions about how to download the service files. Make sure you have access to these. Following is an example.

```
TEXT    = Data sent via "INET". To retrieve your service:
TEXT    =   FTP to: ptf.boulder.ibm.com
TEXT    =   Log on using userid "anonymous" and password "h5q7nep9"
TEXT    =   Enter the following FTP commands:
TEXT    =     cd /390777166/c568411202
TEXT    =     ascii
TEXT    =     get ftp7166.txt
TEXT    =     binary f 1024
TEXT    =     get rlst7166.bin
TEXT    =     get rptf0166.bin
TEXT    =     get rptf0266.bin
TEXT    =     get rptf0366.bin
```

1. Logon to `MAINT`.
2. The `MAINT 500` disk should have a large amount of free space, so it is a useful minidisk on which to download the files. By default, the FTP client saves files on the A disk, so access the 500 disk as A:

```
==> acc 500 a
DMSACC724I 500 replaces A (191)
```

3. Link to the TCPMAINT 592 disk to obtain access to the FTP client command:

```
==> vmlink tcpmaint 592
DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
```

4. Use the FTP client to obtain the RSU envelopes from the Internet. The envelope files can be large, so this may take some time. We recommend that you rename the file type from BIN to SERVLINK through FTP, because this is the file type that the **SERVICE** command expects. As you are downloading the files, note the file sizes. Following is an example.

```
==> ftp ptf.boulder.ibm.com
ftp> <owte8a>
ftp> <h2q9nep9>
ftp> cd </390777166/c568411202>
ftp> ascii
ftp> get ftp7166.txt
ftp> binary f 1024
ftp> get rlst7166.bin rlst8476.servlink
...
10240 bytes transferred in 0.523 seconds. Transfer rate 19.58 Kbytes/sec
ftp> get rptf0166.bin rptf0176.servlink
...
36944896 bytes transferred in 191.632 seconds. Transfer rate 192.79 Kbytes/sec.
ftp> get rptf0266.bin rptf0276.servlink
...
26028032 bytes transferred in 132.353 seconds. Transfer rate 196.66 Kbytes/sec.
ftp> get rptf0366.bin rptf0376.servlink
...
52193280 bytes transferred in 269.094 seconds. Transfer rate 193.96 Kbytes/sec.
ftp> quit
```

You should now have the service or envelope files on your z/VM system.

5. The envelope files arrive in a compressed format to speed downloads. To use them, they must first be uncompressed with the **DETERSE** command. Use the **(REPLACE** parameter to uncompress them in place and save disk space:

```
==> deterse rlst7166 servlink a = = = (replace
==> deterse rptf0166 servlink a = = = (replace
==> deterse rptf0266 servlink a = = = (replace
==> deterse rptf0366 servlink a = = = (replace
```

> **Note:** If you did not rename the file names during the FTP session, you can use the CMS **RENAME** command. Type **HELP RENAME** for help.

6. Use the **BROWSE** command to read the RSU information. Compare the byte count that you recorded earlier with the values in this file.

```
==> browse ftp7166 txt
VM RSU Package Information
--------------------------
This file contains byte counts of files to receive and instructions
for preparing the files for installation. The byte counts listed
below should match the byte counts of the files when they are received
using FTP.

FILE BYTE COUNTS
----------------

The rlst7166.bin byte count is: 10240.
The rptf166.bin byte count is: 36944896.
The rptf266.bin byte count is: 26028032.
The rptf366.bin byte count is: 52193280.
...
```

Archived

5.2.3 Receiving, applying, and building the service

You must receive, apply, and build the service. Then it can be put into production. In the past, this was a cumbersome procedure. For example, to receive, apply and build the CP component, the following steps were needed:

```
vmfmrdsk zvm cp apply (setup
vmfsetup zvm cp
vmfpsu zvm cp
vmfins install ppf zvm cp (nomemo env {filename} nolink override no
vmfapply ppf zvm cp (setup
vmfbld ppf zvm cp (status
vmfbld ppf zvm cp (serviced
```

Then the same steps were needed for many other components. The process is much easier now with the **SERVICE** command. On the other hand, the previous method is more granular and better enables the system administrator to know which pieces of service have been applied.

1. To prepare to use the **SERVICE** command, you must have a 256 MB virtual machine and you must have the minidisk with a large amount of free space; that is what the MAINT 500 minidisk is for. Increase the size of the MAINT virtual machine with the **DEFINE STORAGE** command:

   ```
   ==> def stor 256M
   STORAGE = 256M
   Storage cleared - system reset.
   ==> ipl cms
   IPL CMS
   z/VM V5.4.0    2008-05-13 14:27
   ==> Press Enter
   ```

2. The **SERVICE** command will write to the current A disk, so you again want to access 500 as A:

   ```
   ==> acc 500 a
   DMSACC724I 500 replaces A (191)
   ```

3. Use the **SERVICE ALL** command, specifying the envelope files you downloaded. Many, many screens of output will scroll by and the screens will automatically be cleared. Important messages will be saved to the A (500) disk. This process may take many minutes or tens or tens of minutes. Following is an example:

   ```
   ==> service all rptf0176 rptf0276 rptf0376
   ...
   VMFSUT2760I VMFSUFTB processing completed successfully
   VMFSRV2760I SERVICE processing completed with warnings
   Ready(00004); T=185.66/191.79 11:56:47
   ```

   A return code of zero (0) is ideal. Note in the last Ready line that this command returned a code of 4. In general a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

4. The output files written to the A disk are of the form $VMF* $MSGNUM:

   ```
   ==> filel $VMF* $MSGLOG
   $VMFP2P  $MSGLOG  A1 V         80       1520         28 11/10/05 13:35:43
   $VMFBLD  $MSGLOG  A1 V         80        639          9 11/10/05 13:28:42
   $VMFMRD  $MSGLOG  A1 V         80        499          7 11/10/05 13:28:39
   $VMFSRV  $MSGLOG  A1 V         80       1369         25 11/10/05 11:56:47
   $VMFAPP  $MSGLOG  A1 V         80        682          9 11/10/05 11:54:07
   $VMFINS  $MSGLOG  A1 V         80        381          6 11/10/05 11:54:05
   ```

5. Invoke the **VMFVIEW SERVICE** command to review the results of the previous **SERVICE** command. Following is an example:

```
==> vmfview service
********************************************************************************
****             SERVICE               USERID: MAINT                 ****
********************************************************************************
****            Date: 11/10/05            Time: 11:43:15             ****
********************************************************************************
CK:VMFSUI2104I PTF UM30896 contains user information. Review the :UMEMO
CK:            section in file UM30896 $PTFPART
CK:VMFSUI2104I PTF UM31044 contains user information. Review the :UMEMO
CK:            section in file UM31044 $PTFPART
CK:VMFSUI2104I PTF UM31233 contains user information. Review the :UMEMO
CK:            section in file UM31233 $PTFPART
CK:VMFSUI2104I PTF UM31275 contains user information. Review the :UMEMO
CK:            section in file UM31275 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (I) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTX EXEBIN
WN:VMFSRV1221W The CP Stand-Alone Dump Utility must be rebuilt. Follow
WN:            the instructions in the z/VM Service Guide.
```

   Ideally there will be no output, which means the service applied perfectly. In this example, messages were generated. The first four `VMFSUI2104I` messages are informational. The `VMFBDC2250W` message is pertinent if you are using the VM HCD tool. The `VMFSRV1221W` is pertinent if you are using the CP Stand-alone Dump Utility.

You should see that the service was installed successfully.

5.2.4 Putting the service into production

This section describes how to use the **PUT2PROD** command to put the service into production.

Important: If you run **PUT2PROD** from a 3270 emulator session, you may lose your connection because the TCP/IP service machine may be recycled. Therefore, you may want to run this command from a console. If you do switch to a new session, remember to set the machine size to 256 M and access the 500 disk as A.

1. Use the **PUT2PROD** command to put the service into production. Many screens will scroll by. This command can take quite a number of minutes to complete:

```
==> put2prod
RDR FILE 0016 SENT FROM MAINT    CON WAS 0016 RECS 0004 CPY  001 T NOHOLD NOKEEP
VMFP2P2760I PUT2PROD processing started
...
USER DSC   LOGOFF AS  BLDCMS   USERS = 7     FORCED BY MAINT
VMFP2P2760I PUT2PROD processing completed successfully for SAVECMS
VMFP2P2760I PUT2PROD processing completed successfully
```

Even though the service has been “put into production”, the **QUERY CPLEVEL** command should still return the current service level; in this example, 0501. This is because the new CP load module (nucleus) has not been invoked:

```
==> q cplevel
z/VM Version 5 Release 3.0, service level 0701 (64-bit)
Generated at 05/28/07 23:44:09 EST
IPL at 11/11/07 13:44:37 EST
```

To invoke the new CP load module, use the **SHUTDOWN REIPL** command. When your system comes back up, it should be at the new CP service level; in this example, 0702:

```
==> shutdown reipl iplparms cons=sysc
...
==> q cplevel
z/VM Version 5 Release 3.0, service level 0702 (64-bit)
Generated at 05/29/07 18:39:52 EST
IPL at 11/12/07 14:53:28 EST
```

This shows that the new CP load module is now being used.

5.3 Determining the z/VM service level

Often you will want to be able to query more than just the service level. The following steps were taken from the links CP Maintenance Levels and Virtual Switch TCP/IP Maintenance Levels starting at the Web site:

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

Logon as TCPMAINT. Use the **QUERY VMLAN** command to determine the latest APAR applied:

```
==> cp query vmlan
VMLAN maintenance level:
  Latest Service: VM63895
VMLAN MAC address assignment:
  MACADDR Prefix: 020001
  MACIDRANGE SYSTEM: 000001-FFFFFF
             USER:   000000-000000
VMLAN default accounting status:
  SYSTEM Accounting: OFF       USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 1
  TRANSIENT  Limit: INFINITE   Current: 0
```

This shows that the latest APAR applied is VM63895.

The maintenance level of the TCP/IP stack is important to virtual networking. To determine this, first get the active VSWITCH controller:

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 8    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  State: Ready
  IPTimeout: 5        QueueStorage: 8
  Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1  VDEV:  3004
  Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2  VDEV:  3008 BACKUP
```

This shows the controller is named DTCVSW1. Then 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 530
IBM 2094; z/VM Version 5 Release 3.0, service level 0701 (64-bit), VM TCP/IP Lev
el 530; RSU 0701 running TCPIP MODULE E2 dated 11/10/07 at 11:34
TCP/IP Module Load Address:  00C44000
```

Archived

This shows information about the TCPIP MODULE. Use the **TCPSLVL** command and the complete file specification (TCPIP MODULE E, in this example) to get more information. Of particular interest is the latest APAR applied to TCTOOSD:

```
==> tcpslvl tcpip module e
SLVL  TCPIP     PQ22678
...
SLVL  TCTOOSD   PK00905
...
```

Summary

At this point, you have completed installing, configuring, and servicing z/VM. A valuable attribute of z/VM is that it normally runs with little maintenance required. It is now time to change your focus to Linux.

Archived

Archived

6

Configuring an NFS server for SLES 10 SP2

There are several ways to install SLES 10 SP2 onto the mainframe. It can be installed from FCP-attached CD-ROM (in this case, all data is read directly from CD), or from network sources like NFS, FTP, HTTP and SMB. Installation through an NFS server is used in this book. To accomplish this, it is recommended that you set up a PC Linux system. This server will supply both the SLES 10 distribution and the files associated with this book.

The server must have at least 6 GB of free disk space for one SLES 10 install server. It can be a Linux PC, but it can also be a UNIX machine (Sun™ Solaris™, Hewlett Packard HP-UX, IBM AIX®, or other). The steps in this chapter explain how to configure a Linux PC as the NFS server.

You can also choose to use a Windows workstation through SMB, but this option is not addressed in this book. Often more problems are encountered when using a Windows workstation to serve a SLES 10 SP2 installation, so using this option is not recommended. If you have no other choice, refer to 2.1.2, “Using a Microsoft Windows Workstation” in *SUSE LINUX Enterprise Server ARCHITECTURE SPECIFIC INFORMATION*. This manual is included on the SUSE CDs.

© Copyright IBM Corp. 2008. All rights reserved.

6.1 Getting started

To get started with Linux on System z using this book, perform the following tasks:

- Downloading files associated with this book
- Setting up an SLES 10 SP2 install server
- Enabling the NFS server

These tasks are explained in detail the following sections.

6.2 Downloading files associated with this book

This book has many files associated with it that you will need to set your system up quickly. You can download the tar file on the Web at:

```
ftp://www.redbooks.ibm.com/redbooks/SG247493
```

The tar file `SG24-7493.tgz` is about 16 KB. Download the file and untar it. The following example shows this being done from the directory `/nfs/`:

```
# mkdir /nfs
# cd /nfs
... download or copy the file SG247493.tgz ...
# tar xzf SG247493.tgz
```

This should create a new directory named `virt-cookbook-S10S2/`. List the files:

```
# ls -F virt-cookbook-S10SP2/
README.txt  clone.sh*  vm/
```

There is a README file, a clone script, and a directory for z/VM files. At this point, you now have downloaded and untarred the files associated with this book.

Make a directory `/nfs/sles10sp2/`. This will hold the SLES 10 SP2 install ISO images:

```
# mkdir sles10sp2
```

6.3 Setting up a SLES 10 SP2 install server

You may have a licensed version of SLES 10 SP2 on physical CDs, or you may choose to try an evaluation copy. There is an evaluation copy on the Web starting at the following URL:

```
http://www.novell.com/products/server/eval.html
```

Follow the link named SUSE Linux Enterprise Server 10 SP2 for IBM System z and create an account to download the ISO images.

6.3.1 SLES 10 SP2 DVD ISO image file

Table 6-1 on page 87 lists the DVD ISO images for SLES 10 SP2 that were available at the time of writing.

Archived

Table 6-1 SLES 10 s390x CDs

DVD number	File name	File size in bytes
1	SLES-10-SP2-DVD-s390x-GMC-DVD1.iso	2687717376
2	SLES-10-SP2-DVD-s390x-GMC-DVD2.iso	3277279232

The second DVD is not described in this book because it contains only source RPMs and is not needed.

If you are starting with a soft copy of the DVD 1 ISO image, copy it to the `/nfs/sles10sp2/` directory now.

6.3.2 Starting from physical DVD

Note: Perform the tasks in this section *only* if you are starting from physical DVD discs.

If you are starting with physical CDs or DVDs, you must first convert them to ISO images. This can be accomplished through the Linux **dd** command which basically does a byte-for-byte copy of the CD/DVD contents.

Put the first CD or DVD in the drive. It is often available as the file `/dev/cdrom`. If there is no such file on your system, you will have to determine which file (such as `/dev/hdc`) is the device file for the CD drive.

Now copy the contents of the DVD to an ISO image using the **dd if** (input file) and **of** (output file) parameters. The following example copies the SLES 10 SP2 DVD to the appropriately named file.

Sometimes, `/dev/cdrom` is automatically mounted over `/mnt/cdrom` when you put the CD in the drive. If so, you will need to unmount it through the **umount** command after copying the contents of the CD:

```
# cd /nfs/sles10sp2
# dd if=/dev/cdrom of=SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
# umount /mnt/cdrom
```

The CD or DVD should start spinning and will take several minutes to copy. Repeat the **dd** command for each additional disc you have, using appropriate file names.

6.3.3 Verifying the ISO images

First verify the integrity of the ISO images. This is done through a file of checksum values and ISO file names.The checksums were calculated from the contents of the CD. After downloading or **dd**ing the ISO images, the checksums are calculated again and compared against the original values through the **md5sum** command and the checksum files.

Following is an example of using the **md5sum** command against an MD5SUM file. When you obtain the CDs or DVDs, whether they are physical discs or ISO images, you should also obtain an MD5SUM file.

SLES 10 MD5SUM values

```
# cat MD5SUM
49db05773f590518bd97c81eef3307ff  SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
eb8c72a1a90f78474bf9727b895dba35  SLES-10-SP2-DVD-s390x-GMC-DVD2.iso
```

Use the **md5sum -c** command to verify the integrity of the ISO images. All should report OK. Following is an example:

```
# md5sum -c MD5SUM
SLES-10-SP2-DVD-s390x-GMC-DVD1.iso: OK
SLES-10-SP2-DVD-s390x-GMC-DVD2.iso: OK
```

Any ISO images that do not report OK must be downloaded or copied again.

6.3.4 Configuring the SLES 10 SP2 install server

With SLES 9 and earlier, creating an install tree became complicated with Service Packs. SLES 10 now has the ability to automount the CDs and DVDs. There are different methods of making the SLES 10 SP2 distribution available:

- By copying the contents of all DVDs/CDs into one directory
- By Loop mounting each ISO image to different directories
- By using the ISO images directly

Using the ISO images directly is the easiest option, because the installation process will access the ISO images directly and mount them automatically.

The initial RAMdisk (`initrd`) and the kernel (`vmrdr.ikr`) will have to be copied to z/VM for the first Linux install. These files are on the first DVD (or CD) in the `/boot/` directory.

To allow access to these files, the first DVD must be loopback-mounted using the **-o loop** flag to the **mount** command. First create a directory, `dvd1/`, as a mount point:

```
# cd /nfs/sles10sp2
# mkdir dvd1
# mount -o loop SLES-10-SP2-DVD-s390x-GMC-DVD1.iso dvd1
```

You should now have a directory, `/nfs/`, with subdirectories of `sles10sp2/` and `virt-cookbook-S10SP2/`.

The next step is to enable the NFS server.

6.4 Enabling the NFS server

The method of enabling an NFS server will differ, depending upon the operating system. However, the steps are basically the same:

- Be sure the NFS RPMs are installed.
- Export the appropriate directories.
- Start the NFS server in the current run level.

First, be sure the NFS server is installed. Typically, the RPM is named `nfs-utils`. If this RPM is not installed, then install it now.

The directories to export through NFS are set in the `/etc/exports` configuration file. Make a backup copy of the file. Then edit the original copy and add the two directories to be exported:

```
# cd /etc
# cp exports exports.orig
# vi exports       // add two lines at the bottom
/nfs/virt-cookbook-S10SP2       *(ro,sync)
/nfs/sles10sp2                  *(ro,sync)
```

The *(ro,sync) parameter specifies that any client with access to this server can get the NFS mount read-only. You may want to be more restrictive than any client (*) for security reasons. Type: man exports for more details.

Also be sure that the NFS server is running in your run level; for a SLES Linux, the service name is nfsserver. Use the **chkconfig --list** command:

```
# chkconfig --list nfsserver
nfsserver               0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

The output shows that the NFS server is set up to run in the most common run levels: 3 and 5.

If your NFS server is not set to start, you will need to set it to run with the **chkconfig** command and turn it on for the current run level with the **rcnfsserver start** command (if the NFS server is already running, you can use the **restart** parameter):

```
# chkconfig nfsserver on
# rcnfsserver start
Starting kernel based NFS server
done
```

Your NFS server should now be running with the directory exported. We recommend that you test this by mounting the exported directory locally. The following example shows that the /mnt/ directory is empty. Then the newly exported /nfs/ directory is mounted using the localhost keyword, and the files are listed.

```
# ls /mnt
# mount localhost:/nfs/sles10sp2 /mnt
# ls -F /mnt
SLES-10-SP2-DVD-s390x-GMC-DVD1.iso  SLES-10-SP2-DVD-s390x-GMC-DVD2.iso
```

This shows that the SLES 10 install root directory is accessible. Now unmount it and test the virt-cookbook-S10PS2/ directory:

```
# umount /mnt
# mount localhost:/nfs/virt-cookbook-S10SP2 /mnt
# ls -F /mnt
README.txt  clone.sh*  vm/
# umount /mnt
```

At this point, you should be able to use this server as the source of your first mainframe Linux installation. Later you will be able to copy the install tree and keep it on System z Linux.

Archived

Archived

7

Installing SLES10 SP2 on the controller

Note: Chapters Chapter 4, “Installing and configuring z/VM” on page 27, 5, and Chapter 6, “Configuring an NFS server for SLES 10 SP2” on page 85 must be completed before you proceed with this chapter.

This chapter describes how to install SLES 10 SP2 onto the user ID `LINUX00`, which is referred to as the *controller*. It explains how to configure the controller to provide the following function:

- A SLES10 SP2 install server - a tree of the RPMs and other files needed for installation
- Cloner - the ability to copy a Linux system from a logged-off source user ID to a target user ID with identical minidisks.
- NFS server - to export the *install tree* and possibly other data and directories
- Time server (optional) - for time synchronization of other Linux virtual servers. (To make the controller a time server, see 11.4, “Setting the software clock through NTP” on page 192.)

The following steps are detailed:

- Creating the user ID LINUX00
- Updating AUTOLOG1 PROFILE EXEC
- Preparing SLES10 SP2 bootstrap files
- Installing the controller
- Configuring the controller

© Copyright IBM Corp. 2008. All rights reserved.

7.1 Creating the user ID LINUX00

In this section you will define the user ID LINUX00. Think of this as defining a virtual PC with memory and small disks and a Network Interface Card (NIC), but no operating system (at this point).

To accomplish this, perform the following steps:

1. Logon to MAINT.
2. Edit the USER DIRECT file:

   ```
   ==> x user direct c
   ```

   In the USER DIRECT file, you can group statements that will be common to many user definitions in a construct called a *profile*. This profile can then become part of the user definitions through the INCLUDE statement. You used the existing profile TCPCMSU when you defined the LNXMAINT user.
3. Create a new profile named LNXDFLT. This will contain the user directory statements that will be common to all Linux user IDs. To save typing, you can use the **""** prefix commands to duplicate the IBMDFLT profile that should be on lines 37-50:

   ```
   ""037 ************************************************************
   00038 *
   00039 PROFILE IBMDFLT
   00040   SPOOL 000C 2540 READER *
   00041   SPOOL 000D 2540 PUNCH A
   00042   SPOOL 000E 1403 A
   00043   CONSOLE 009 3215 T
   00044   LINK MAINT 0190 0190 RR
   00045   LINK MAINT 019D 019D RR
   00046   LINK MAINT 019E 019E RR
   00047   LINK MAINT 0402 0402 RR
   00048   LINK MAINT 0401 0401 RR
   00049   LINK MAINT 0405 0405 RR
   ""050 *
   ```

4. Edit the duplicated profile by deleting the three LINK MAINT 040x lines, and inserting the lines that are in bold text below:

   ```
   PROFILE LNXDFLT
     IPL CMS                                  1
     MACHINE ESA 4                            2
     CPU 00 BASE                              3
     NICDEF 600 TYPE QDIO LAN SYSTEM VSW1     4
     SPOOL 000C 2540 READER *
     SPOOL 000D 2540 PUNCH A
     SPOOL 000E 1403 A
     CONSOLE 009 3215 T
     LINK MAINT 0190 0190 RR
     LINK MAINT 019D 019D RR
     LINK MAINT 019E 019E RR
     LINK LNXMAINT 192 191 RR                 5
     LINK TCPMAINT 592 592 RR                 6
   ```

   Notes:

   1 CMS will be IPLed when the user ID is logged onto.
   2 Machine will be of type ESA with a maximum of 4 CPUs that can be defined.
   3 Defines the base CPU (Some workloads will benefit from a second virtual CPU by adding another line with CPU 01.)

4 Defines a virtual NIC connected to the VSWITCH starting at virtual address 600.
5 Provides read access to the LNXMAINT 192 disk as the user's 191 disk.
6 Provides read access to the TCPMAINT 592 disk, so that the user has access to TCPIP services such as an FTP client.

5. Go to the bottom of the file and add the definition for a new user ID named LINUX00. This user ID is given class B privilege in order to run the **FLASHCOPY** command; class D privilege to run the **QUERY ALLOC MAP** command; and class E privilege to run the **QUERY NSS** command. Be sure to replace the volume labels shown in this example with the labels of your DASD:

```
*
USER LINUX00  LNX4VM 512M 1G BDEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3338 <MMD856> MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 0321 3018 <MMD857> MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 0001 3338 <MMD952> MR LNX4VM LNX4VM LNX4VM
 MDISK 103 3390 0001 3338 <MMD953> MR LNX4VM LNX4VM LNX4VM
 MDISK 104 3390 0001 3338 <MMD954> MR LNX4VM LNX4VM LNX4VM
```

This Linux user ID will have the following minidisks:

100 A single 3390-3 for the root file system.

101-104 4 3390-3s that will be used to create a logical volume mounted over /nfs/. This will store the SLES10 SP2 installation tree and the files associated with this book available through NFS.

Note that the 3390 and address D857 starts at cylinder 321 because the first 320 cylinders were used for the LNXMAINT 191 and 192 disks.

6. Go back to the top of the file and search for string USER $ALLOC$. Add cylinder 0 of each of the four new volumes to this dummy user ID so they do not show up as gaps.

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 <MVD850> R
 MDISK A02 3390 000 001 <MVD853> R
 MDISK A03 3390 000 001 <MVD854> R
 MDISK A04 3390 000 001 <MMD857> R
 MDISK A05 3390 000 001 <MMD856> R
 MDISK A06 3390 000 001 <MMD952> R
 MDISK A07 3390 000 001 <MMD953> R
 MDISK A08 3390 000 001 <MMD954> R

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

7. Run **DISKMAP** to check for overlaps and gaps. You should only see the two gaps that are inherent in the default USER DIRECT file. Press **F3** to quit.

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  331  line(s) not displayed --------------------
====> F3
```

8. When the disk layout is correct, run **DIRECTXA** to bring the changes online:

```
==> directxa user
```

```
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 44 disk pages
```

At this point, you have defined the user ID that will become the controller.

7.2 Updating AUTOLOG1 PROFILE EXEC

The new Linux user ID, LINUX00, will need access to the VSWITCH. A **SET VSWITCH** command with the **GRANT** parameter can be added to AUTOLOG1's **PROFILE EXEC** to do this. Also, an **XAUTOLOG** statement can be added so the controller is automatically logged on at z/VM IPL time:

Link and access the AUTOLOG1 191 disk read-write and edit the file PROFILE EXEC. Add LINUX00 to the section that grants access to the VSWITCH. Also add LINUX00 to the section that uses **XAUTOLOG** to automatically log on the Linux user IDs:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
==> x profile exec f
/*************************/
/*  Autolog1 Profile Exec  */
/*************************/
'cp xautolog tcpip'                 /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'        /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant linux00'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog linux00'
'cp logoff'                          /* logoff when done */
====> file
```

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

```
==> set vswitch vsw1 grant linux00
Command complete
```

The new user ID, LINUX00, has now been granted access to the VSWITCH VSW1 for this session and at IPL time, and it automatically logs on at IPL time.

7.3 Preparing SLES10 SP2 bootstrap files

To IPL a SLES10 SP2 installation system, three bootstrap files must be copied and "punched" to the reader. Then you can IPL an install system from the reader (virtual address

00C). The three files are a kernel, a parameter file, and an initial RAMdisk. Think of these files as a PC Linux boot floppy or CD. Also, a small REXX EXEC is commonly used to clean out the reader, punch the three files, and IPL the reader.

1. Start an SSH session as root on the NFS server.
2. Change directory to the mounted DVD. You should see a directory boot/ where the kernel and RAMdisk are located:

```
# cd /nfs/sles10sp2/dvd1
# ls
ARCHIVES.gz   README          docu/                            ls-lR.gz
COPYING       README.BETA     dosutils/                        media.1/
COPYING.de    boot/           gpg-pubkey-0dfb3188-41ed929b.asc patches/
COPYRIGHT     content         gpg-pubkey-307e3d54-44201d5d.asc pubring.gpg
COPYRIGHT.de  content.asc     gpg-pubkey-3d25d3d9-36e12d04.asc suse/
ChangeLog     content.key     gpg-pubkey-7e2e3b05-44748aba.asc suse.ins
INDEX.gz      control.xml     gpg-pubkey-9c800aca-40d8063e.asc
NEW           directory.yast  gpg-pubkey-a1912208-446a0899.asc
```

3. Change directory to boot/s390x/ and invoke the **ls** command. You should see the initial RAMdisk and kernel named **initrd** and **vmrdr.ikr**.

```
# cd boot/s390x
# ls
cd.ikr          initrd      parmfile     rescue  root.fonts  tapeipl.ikr
directory.yast  initrd.siz  parmfile.cd  root    suse.ins    vmrdr.ikr
```

4. FTP to z/VM and login as LNXMAINT. Copy the SLES10 SP2 kernel (the file vmrdr.ikr copied as SLES10S2 KERNEL) and the initial RAMdisk (the file initrd copied as SLES10S2 INITRD). These files must have a record format of fixed 80 byte records and be transferred in binary. This format can be set with the **site fix 80** FTP subcommand (if this subcommand fails, try **quote site fix 80**).

   Following is an example:

```
# ftp <9.12.5.22>
Connected to 9.12.5.22.
Name (9.12.5.22:root): lnxmaint
331-Password: lnx4vm
230-220-FTPSERVE IBM VM Level 540 at VMLINUXA.ITSO.IBM.COM, 13:53:35 EDT FRIDAY
2008-05-16
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> bin
200 Representation type is IMAGE.
ftp> site fix 80
200 Site command was accepted.
ftp> put vmrdr.ikr SLES10S2.KERNEL
local: vmrdr.ikr remote: SLES10.KERNEL
...
ftp> put initrd SLES10S2.INITRD
local: initrd remote: SLES10S2.INITRD
...
ftp> quit
```

5. Go back to your 3270 session. Logoff MAINT and logon to LNXMAINT.
6. In addition to the kernel and RAMdisk that you just copied, the file SLES10S2 EXEC should exist on the LNXMAINT 192 disk.

   Use the **FILELIST** command to verify that the kernel and RAMdisk were copied in fixed-80 byte record format.

You should see the following files:

```
==> filel sles10s2 * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=18 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SLES10S2 EXEC     D1 V         69          9          1  5/16/08 15:24:52
      SLES10S2 INITRD   D1 F         80     106644       2083  5/16/08 14:18:40
      SLES10S2 KERNEL   D1 F         80      89413       1229  5/16/08 14:18:27
```

7. Quit by pressing **F3**.
8. Verify that the file **SLES10S2 EXEC** has the correct information. Note the kernel and RAMdisk have hardcoded file names, but the file name of the parameter file will be the user ID (**userid()** function) of the user running the EXEC:

```
==> type sles10s2 exec d
/* EXEC to punch SLES10S2 install system to reader and IPL from it */
'CP SPOOL PUN *'
'CP CLOSE RDR'
'PUR RDR ALL'
'PUN SLES10S2       KERNEL * (NOH'
'PUN' userid()   'PARM-S10 * (NOH'
'PUN SLES10S2       INITRD * (NOH'
'CH RDR ALL KEEP'
'IPL 00C CLEAR'
```

9. A sample parameter file named SAMPLE PARM-S10 is provided to save typing. These are the parameters, especially networking values, that will be used to install SLES. Observe the contents through the **TYPE** command.
10. View the SLES 10 sample parameter file through the **TYPE** command:

```
==> type sample parm-s10 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=n.n.n.n Hostname=xxxxxx.xxx.xxx
Gateway=n.n.n.n Netmask=n.n.n.n
Broadcast=n.n.n.n Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=n.n.n.n Portname=dontcare
Install=nfs://n.n.n.n/nfs/sles10s2/SLES-10-SP2-DVD-s390x-GMC-CD1.iso
UseVNC=1  VNCPassword=xxxxxx
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

11. Copy the sample parameter file to a new file with the file name LINUX00. Edit the file and update the networking variables with the values correct for your site. Refer to the worksheet in 2.7.3, “Linux resources worksheet” on page 18. The fields you should change are in <bold>. The examples used in this book are as follows:

```
==> copy sample parm-s10 d linux00 = =
==> x linux00 parm-s10 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=<9.12.5.30> Hostname=<virtc530.itso.ibm.com>
Gateway=<9.12.4.1> Netmask=<255.255.252.0>
Broadcast=<9.12.4.255> Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=<9.12.6.7> Portname=dontcare
Install=nfs://<9.12.4.185>/nfs/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
UseVNC=1  VNCPassword=<lnx4vm>
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

12. Save your changes with the **FILE** subcommand.

Note: The `Hostname` parameter is not used by the install process; however, it is used in the **`clone.sh`** script that also reads these parameter files.

Now you are ready to start the controller installation.

7.4 Installing the controller

In this section you will install the controller Linux image onto the new user ID `LINUX00`.

1. Logon to `LINUX00`, the controller user ID. You should see a virtual NIC being created at virtual address 600, and the common `PROFILE EXEC` running, which creates two VDISKS for swap spaces at virtual addresses 300 and 301:

```
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 13:52:03 EDT MONDAY 05/19/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
```

2. Answer no (`n`) to the question asking you to IPL Linux from 100:

```
Do you want to IPL Linux from minidisk 100? y/n
n
```

3. You may want to look the `PROFILE EXEC` through the **`TYPE`** command:

```
==> type profile exec
/* 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 /* else */
```

4. Run the **`SLES10S2 EXEC`** to purge the reader, punch the bootstrap files and IPL from the reader. You should see the Linux RAMdisk getting loaded into memory:

```
==> sles10s2
00:      NO FILES PURGED
00: RDR FILE 0001 SENT FROM LINUX00  PUN WAS 0001 RECS 089K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0002 SENT FROM LINUX00  PUN WAS 0002 RECS 0044 CPY  001 A NOHOLD NO
```

```
KEEP
00: RDR FILE 0003 SENT FROM LINUX00  PUN WAS 0003 RECS 107K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Linux version 2.6.16.60-0.17-default (geeko@buildhost) (gcc version 4.1.2 200701
15 (SUSE Linux)) #1 SMP Tue Apr 22 07:28:35 UTC 2008
We are running under VM (64 bit mode)
...
```

Important: If you see the following output on your 3270 window, there is a problem.

```
Main Menu

1) Settings
2) System Information
3) Kernel Modules (Hardware Drivers)
4) Start Installation or System
5) Verify Installation CD-ROM/DVD
6) Eject CD
7) Exit or Reboot
8) Power off
```

Often, this is received because the install program cannot successfully mount the NFS file system, or it can mount it, but cannot find the SLES 10 SP2 install tree. Also, this can result when there is not enough memory (though the example uses 512 MB, which should be plenty).

If you get these messages, you may want to immediately restart the install process with the **IPL 00C** command which IPLs from the reader. Watch the console messages carefully.

5. A VNC server process will be started. Start a new VNC viewer session. You should see the messages:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.12.5.30, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.12.5.30:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

NET: Registered protocol family 10
lo: Disabled Privacy Extensions
IPv6 over IPv4 tunneling driver
*** Starting YaST2 ***
Xlib:  extension "XInputExtension" missing on display ":0.0".
```

6. From your workstation, you can open a Java™-enabled browser to access YaST2 at the specified URL. The logon prompt in Figure 7-1 on page 99 shows VNC access through a Java-enabled browser.

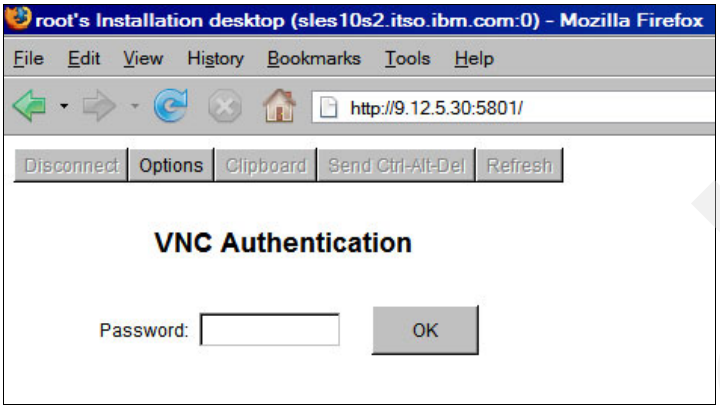


Figure 7-1 VNC viewer through a Java-enabled browser

In addition to a browser, you can also use a standalone VNC viewer if desired (RealVNC, in this example). The server to connect to would be `9.12.5.30:1` in this example, as shown on the left side of Figure 7-2. Enter the password specified in the parameter file (`lnx4vm`, in this example).

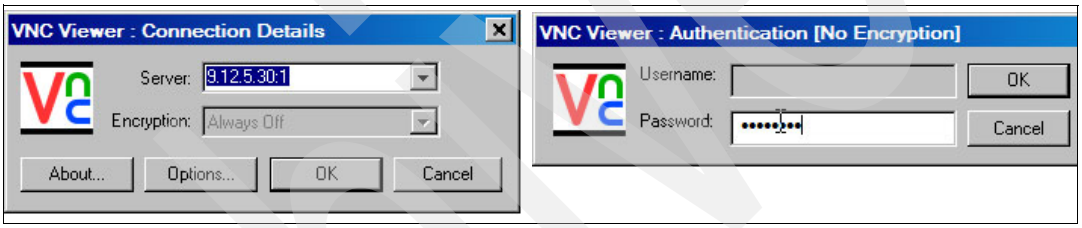


Figure 7-2 Using the VNC viewer

7. You should see a window entitled Language. Select your Language and click **Next**.
8. At the License Agreement window, accept the license agreement and click **Next**.
9. At the Disk Activation window click the **Configure DASD Disks** button.
10. At the DASD Disk Management window:
    a. Select all seven read-write disks (**100-104**, **300** and **301**).
    b. Click **Activate** as shown on the left side of Figure 7-3 on page 100. The DASD will be activated quickly.
    c. Deselect **300** and **301** so that five disks, 100-104, are selected.
    d. Click **Format** as shown on the right side of Figure 7-3 on page 100.
    e. Click **OK** to the query to format 5 parallel disks and then **Yes** to confirm. The DASD will be formatted in parallel. This will take a number of minutes.
    f. Click **Next** when the formatting is complete.
    g. In the Disk Activation window, click **Next**.

Archived

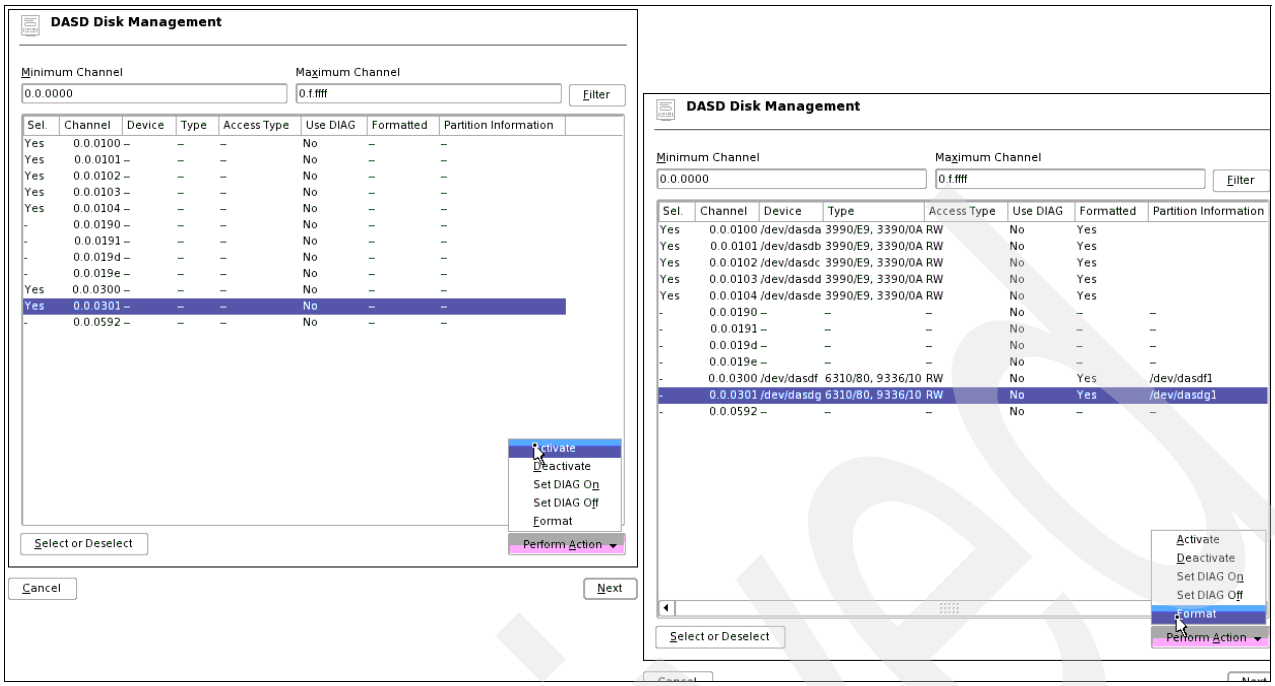


Figure 7-3 Activating and formatting DASD on the controller

11. In the Installation Mode window, accept the default of **New installation** and Click **Next**.
12. In the *Clock* and Time Zone window, select your time settings and click **Next**.
13. In the Installation Settings window, click **Partitioning** and the Expert Partitioner window will appear.
14. Click **Create**. The window Disk to Partition should appear.
15. In the Disk to Partition window, select **/dev/dasda** and click **OK**. The window Create a Primary Partition on /dev/dasda should appear.
16. Set the file system type to **ext3**. Accept the default of using every cylinder and a mount point of / (the root file system) as shown on the left side of Figure 7-4 on page 101.
17. Click the **Fstab Options** button. In the Fstab options window, select the **Device path** radio button in the `Mount in /etc/fstab by` radio group as shown on the right side of Figure 7-4 on page 101. This is so the path /dev/disk/by-path/ccw-0.0.100 will be set in the `/etc/fstab` file, not the disk ID. Click **OK**.
18. Click **OK** in the Create a Primary Partition on /dev/dasda window.

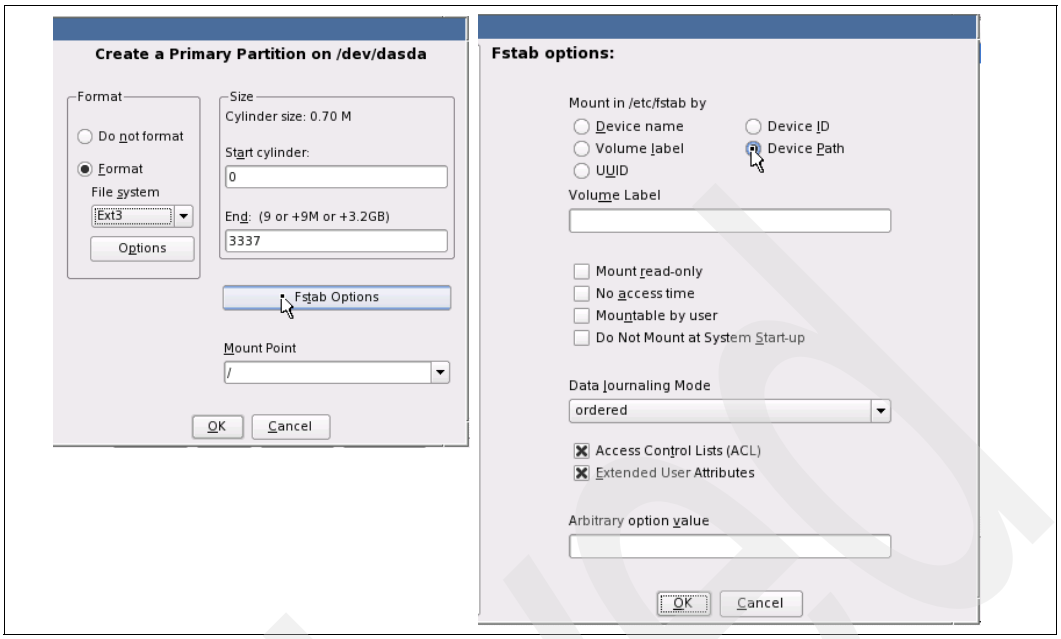


Figure 7-4 Creating a primary partition with /etc/fstab mounting by device path

19. Back on the Expert Partitioner window, click the **Create** button again.
20. On the Disk to Partition window, select **/dev/dasdb** as shown on the left side of Figure 7-5 and click **OK**.
21. On the Create a Primary Partition window, you will create an empty partition, `/dev/dasdb1`, to be used in a logical volume. Click the **Do not format** radio button and select a blank Mount Point (bottom-most entry) as shown on the right side of Figure 7-5. Click **OK**.

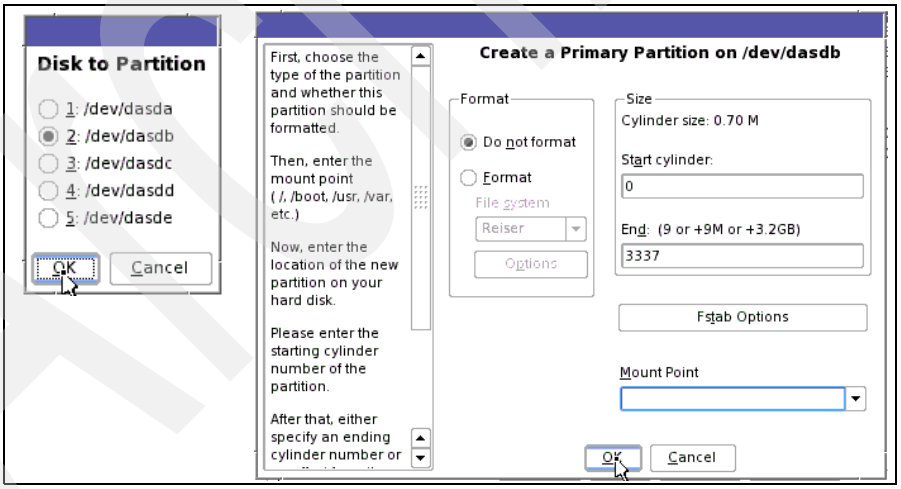


Figure 7-5 Partitioning /dev/dasdd to mount over /backup/

22. You should now see a `/dev/dasdb1` in the Expert Partitioner window. **Repeat** the previous two steps for `/dev/dasdc`, `/dev/dasdd` and `/dev/dasde`. This will give you four partitions `/dev/dasdb1-dasde1` for use with LVM.

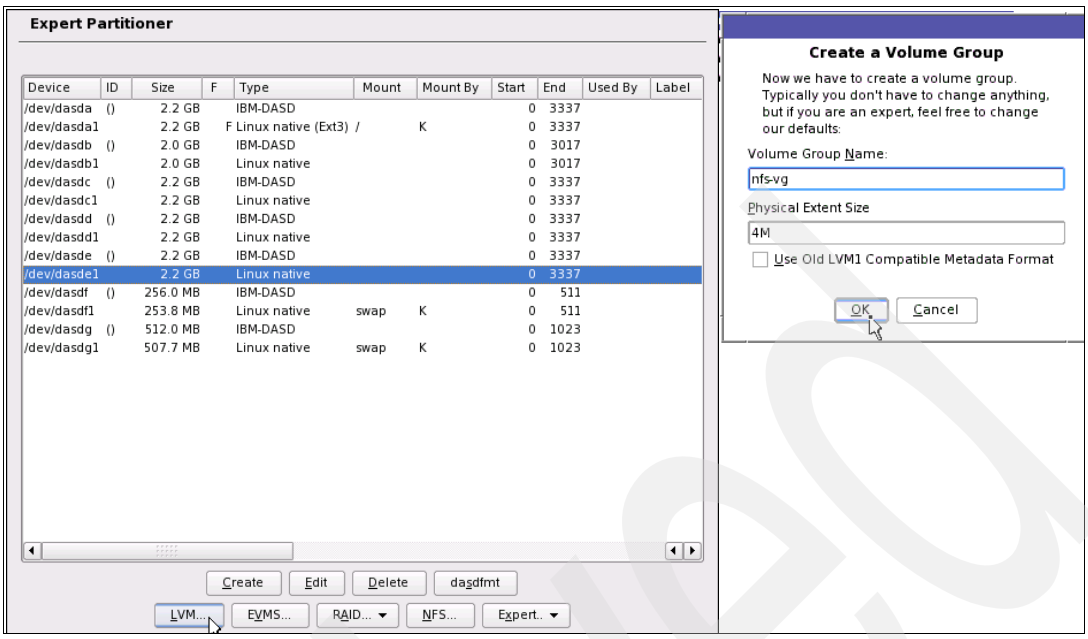


Figure 7-6 Creating a volume group

23. In the Expert Partitioner window, click the **LVM** button as shown on the left side of Figure 7-6. This will start the process of creating a logical volume.
24. In the resulting Create a Volume Group window as shown on the right side of Figure 7-6, set the volume group name to **nfs-vg**. Accept the default Physical Extent Size of **4M**. Click **OK.**
25. In the Physical Volume Setup window as shown in Figure 7-7 on page 103, select **/dev/dasdb1** and click the **Add Volume** button.
26. Repeat the previous step to add the three remaining volumes **/dev/dasdc1**, **/dev/dasdd1** and **/dev/dasde1**. This should create a volume group of size 8.9 GB. Click **Next**.

Archive

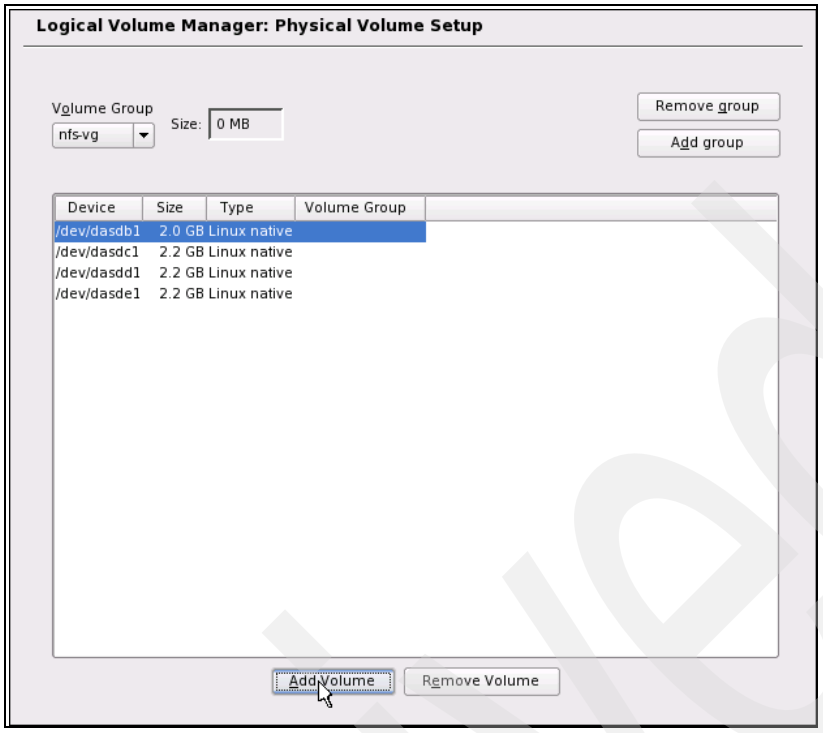


Figure 7-7 Physical Volume Setup

27. In the Logical Volumes window as shown in Figure 7-8, click **Add**. This will allow you to create a single logical volume from the volume group that you just created.

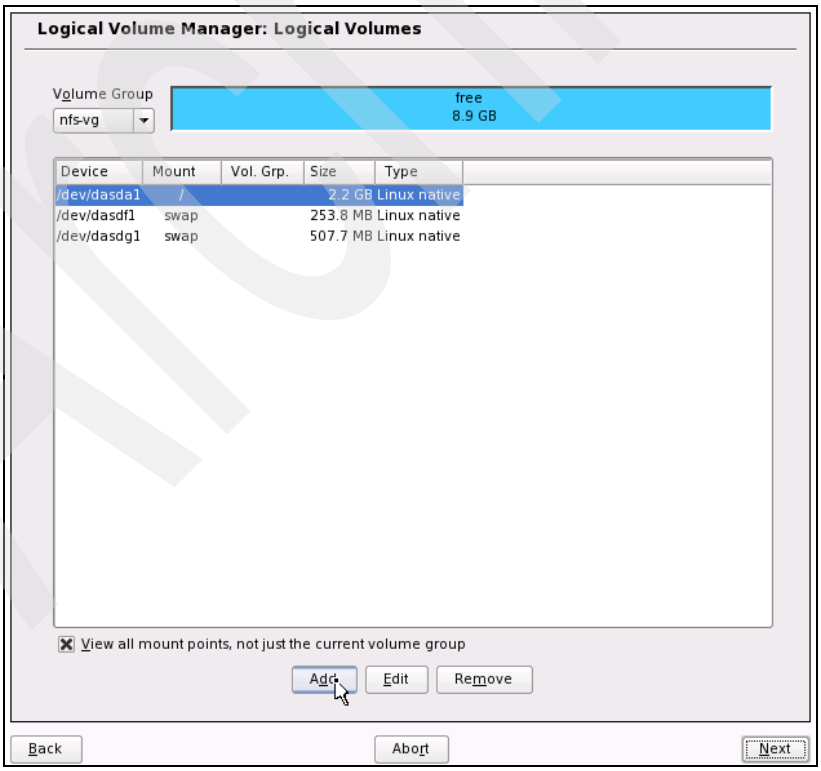


Figure 7-8 Logical Volumes window

28. You should see the Create Logical Volume window as shown in Figure 7-9. Set the Logical volume name to **nfs-lv**. Click the button **max** so it gets the full 8.9 GB. Select the **Format** radio button. Choose a File system type of **ext2**. Specify a Mount Point of `/nfs`. Click **OK**.

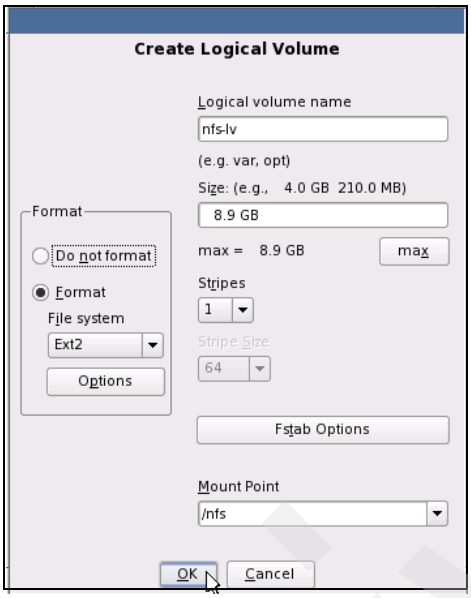


Figure 7-9 Create Logical Volume window

29. Back in the Logical Volumes window, click **Next**.
30. Double-click the swap space **/dev/dasdf1**. The Edit Existing Partition /dev/dasdf1 window will appear.
31. Click the **Fstab options** button as you did previously. The Fstab options window will appear.
32. Select the **Device Path** radio button in the Mount in /etc/fstab by radio group. Click **OK**.
33. In the Edit Existing Partition /dev/dasdf1 window, click **OK**.
34. Repeat the previous steps for the second swap space, **/dev/dasdg1**.
35. You should see a summary as shown in Figure 7-10 on page 105.

Archived

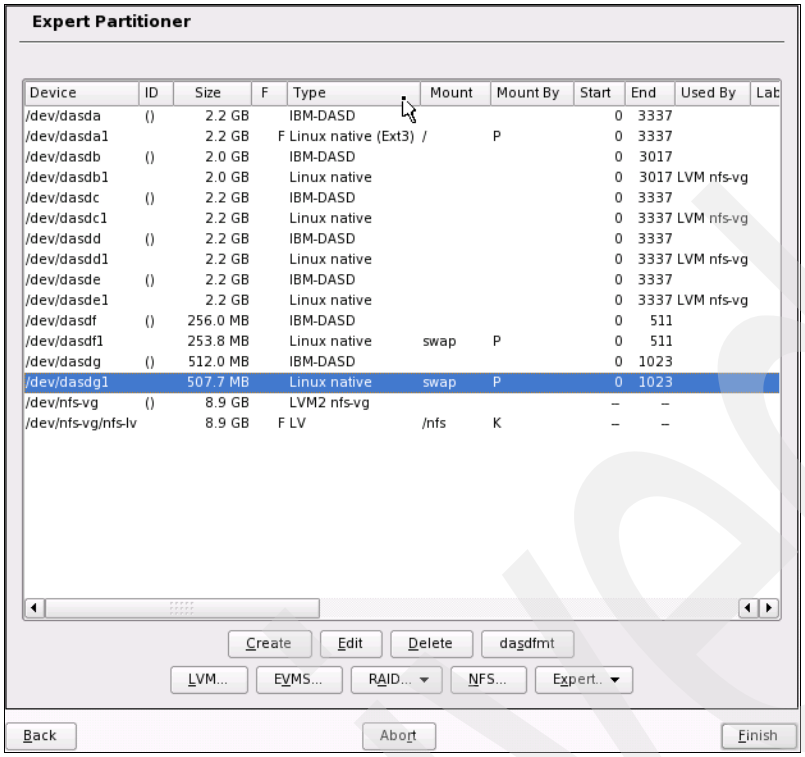


Figure 7-10 Partitioning summary

36. Click **Finish** in the Expert Partitioner window.

37. Back on the Installation Settings window, click **Software**. Deselect all items except **Server Base System**, as shown in Figure 7-11 on page 106. To deselect completely, you must click the box twice. Click **Accept**.

Archive

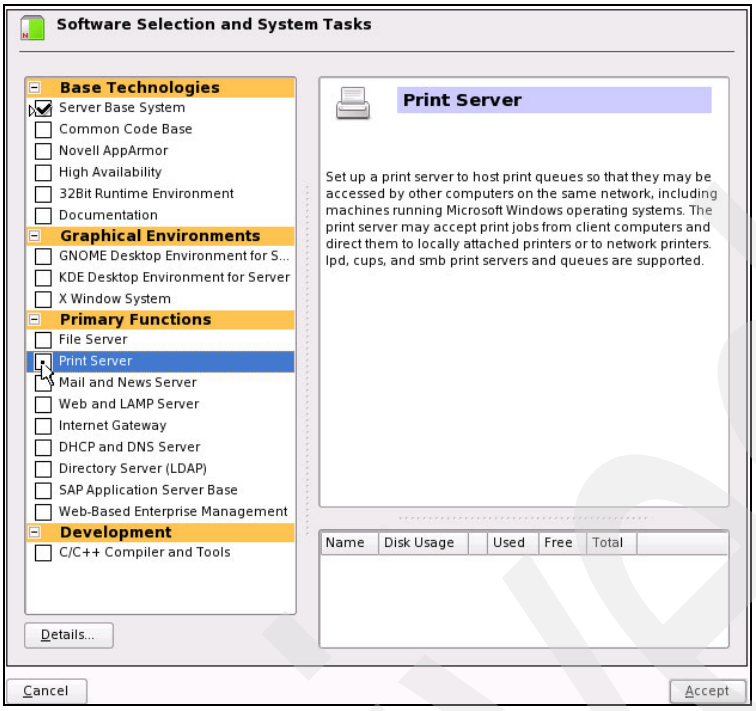


Figure 7-11 Choosing software groups

38. In the Installation Settings window, click **Accept**.

39. On the Confirm Installation window, click **Install**. This will begin the process of laying down RPMs onto your system. Copying the RPMs should take about 5 to 20 minutes depending on network and disk speeds. When the RPM copying is done, your VNC viewer session will stop and may close.

Completing the controller installation

1. Go back to your 3270 session. After the initial system is shut down, you must IPL from minidisk 100. This will boot your new Linux system from disk rather than the install system from the reader.

```
...
01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
00: HCPGSP2630I The virtual machine is placed in CP mode due to a SIGP stop and
store status from CPU 00.
==> IPL 100
00: zIPL v1.6.3 interactive boot menu
00:
00:  0. default (ipl)
00:
00:  1. ipl
00:  2. Failsafe
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 10 seconds):
00: Booting default (ipl)...
...
***
***           You can connect to 9.12.5.30, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.12.5.30:5801/
```

```
***

(When YaST2 is finished, close your VNC viewer and return to this window.)
...
```

2. Start the VNC viewer session again.
3. On the Password for root user window, set the root password and click **Next**.
4. On the Host and Domain Name window, set the Hostname and Domain Name fields appropriately. **Uncheck** the box Change Hostname through DHCP. Click **Next**.
5. On the Network Configuration window in the Firewall section, click **enabled** which will disable the firewall. Click **Next**.
6. On the Test Internet Connection window, if you do not have an Internet connection, select **No, Skip This Test** and click **Next**. (If you have an Internet connection, we recommend that you accept the default).
7. On the Installation Settings window, accept the defaults and click **Next**. A certificate will be created.
8. On the User Authentication Method window, accept the default of **Local (/etc/passwd)** and click **Next**.
9. On the Add a new local user window, add a non-root user and click **Next**.
10. At the Release Notes window, we recommend that you read the information. Click **Next**.
11. At the Hardware Configuration window, click **Next**.
12. At the Installation Complete window, deselect the check box click **Clone this system for Autoyast**, then click **Finish**.
13. The VNC viewer session will end. Go back to the 3270 session and **DISCONNECT** using the **#CP** prefix. You will be able to access the new system through SSH:

```
==> #cp disc
```

You have now installed the controller.

7.4.1 Verifying the installation

Start a new SSH session to the controller as root. Verify some settings through the **df -h** and **swapon -s** commands. You see the following file systems mounted and the two swap spaces:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  1.3G  866M  61% /
udev                  247M  108K  247M   1% /dev
/dev/mapper/nfs--vg-nfs--lv
                      8.8G  149M  8.2G   2% /nfs
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdf1                             partition       259956  0       -1
/dev/dasdg1                             partition       519924  0       -2
```

Linux is now installed on the controller. The next step is to configure it.

7.5 Configuring the controller

Now that your controller is installed, it must be configured. The following steps are involved:

- Copying files to the controller
- Resetting the install source location
- Configuring the NFS server
- Turning off unneeded services
- Applying service if necessary - online update
- Installing the cmsfs package
- Enabling the vmcp and cmm modules
- Setting system to halt on SIGNAL SHUTDOWN
- Rebooting the system
- Verifying the changes

7.5.1 Copying files to the controller

The controller can now be configured to replace the NFS (PC) server to make the SLES10 SP2 install tree and the files associated with this book available through NFS. Copy the following two files from the NFS server to the controller.

- The SLES 10 SP2 DVD ISO image
- The tar file associated with this book: SG24-7493.tgz

Perform the following steps:

1. Create the directory `/nfs/sles10sp2/`:

```
# cd /nfs
# mkdir sles10sp2
```

2. Change directory into the `sles10sp2/` directory and copy the DVD ISO image(s) from the PC NFS server to this directory.

```
# cd sles10sp2
# scp <9.12.4.185>:/nfs/sles10sp2/*.iso .
The authenticity of host '9.12.4.185 (9.12.4.185)' can't be established.
RSA key fingerprint is 65:40:31:d0:fb:db:3e:ef:7d:e7:6e:b5:38:51:35:47.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.4.185' (RSA) to the list of known hosts.
Password:
SLES-10-SP2-DVD-s390x-GMC-DVD1.iso            3%   97MB  13.9MB/s   02:56 ETA
```

3. Change directory to `/nfs/` and copy the files associated with this book from the PC NFS server to this directory:

```
# cd /nfs
# scp <9.12.4.185>:/nfs/SG24-7493.tgz .
```

4. Untar the file `SG24-7493.tgz`:

```
# tar xzf SG24-7493.tgz
```

5. Change into the newly-created directory `virt-cookbook-S10SP2/` and look at the files:

```
# cd virt-cookbook-S10SP2/
# ls -F
README.txt  clone.sh*  vm/
```

6. Copy the script **clone.sh** file to `/usr/local/sbin/`. It will be used later for cloning:

```
# cp clone.sh /usr/local/sbin
```

You have now copied the files you will need from the PC NFS server.

7.5.2 Resetting the install source location

When SLES 10 is installed, the location of the *install source* is remembered. In this case, it is the PC NFS server. Now that the DVD ISO image has been copied from the PC NFS server to the controller, you can reset the install source location to point to `/nfs/sles10sp2` in the local file system. To perform this task, follow these steps:

1. Invoke yast:

   ```
   # yast
   ```

2. Accept the default of `Software` in the left column and select **Installation Source** in the right column and press **Enter**.

3. At the top of the Configured Software Catalogs window, you should see a single entry for SLES 10 SP2. Add a new installation source by using the Tab key to move to the [Add] button at the bottom and pressing **Enter**.

   ```
   Configured Software Catalogs
   +-------------------------------------------------+
   ¦Status¦Refresh¦Name                              ¦
   ¦On    ¦On     ¦SUSE Linux Enterprise Server 10 SP2¦
   ...
   ...
   [Add][Edit][Delete]                [Source Settingsv]

                            [Abort]              [Finish]
   ```

4. Use the Tab key to move to **Local Directory...** and press **Enter** to select. Move to the Next field and press **Enter**.

   ```
   ( ) Scan Using SLP...
   ( ) FTP...
   ( ) HTTP...
   ( ) HTTPS...
   ( ) SMB/CIFS
   ( ) NFS...
   ( ) CD...
   ( ) DVD...
   (x) Local Directory...
   ( ) Package Directory...
   ( ) Specify URL...
   ```

5. Set the Path to Directory or ISO image to `/nfs/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso`.

   Move to the ISO Image field and press **Enter** to select. Move to Next and press **Enter**:

   ```
   ¦ Path to Directory or ISO Image
   ¦ <sles10sp2/SLES-10-SP2-s390x-GMC-DVD1.iso_[Browse...]
   ¦ [x] ISO Image
   ```

6. In the License Agreement window, move to `Yes` and press **Enter** to select. Move to Next and press **Enter**:

   ```
   (x) Yes, I Agree to the License Agreement
   ( ) No, I Do Not Agree
   ```

7. You should now see the Configured Software Catalogs window again, this time with two entries. It is not clear from the content, but the second entry is the one that was just added. Unless you know you are using ZENworks, use the Tab key to move to Synchronize Changes with ZENworks and press **Enter** to deselect it.

Archived

8. With the first entry highlighted (a brighter color), use the Tab key to move to [Delete] and press **Enter**:

```
Configured Software Catalogs
 +-------------------------------------------------+
 ¦Status¦Refresh¦Name                              ¦
 ¦On    ¦On     ¦SUSE Linux Enterprise Server 10 SP2¦
 ¦On    ¦On     ¦SUSE Linux Enterprise Server 10 SP2¦
  ...

 [ ] Synchronize Changes with ZENworks
 [Add][Edit][Delete]               [Source Settingsv]

 [Abort]              [Finish]
```

9. Confirm the delete by choosing `Yes`.

10. Use the tab key to move to [Finish] and press **Enter**.

The controller should now be pointing to the local directory `/nfs/sles10sp2/` as the source of RPMs. You may want to test this by selecting **Software => Software Management** on the YaST Control Center window and moving to the Search button and searching for some software (for example, **xterm**).

7.5.3 Configuring the NFS server

Configure the NFS server to make the SLES 10 SP2 install directory available through NFS. First add the directory to the `/etc/exports` file:

```
# cd /etc
# vi exports
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.
/nfs/sles10sp2          *(ro,sync)
```

The NFS server will be used to export the SLES10 SP2 install tree. The two services needed are named **nfsboot** and **nfsserver**. Turn on these services through the **chkconfig** command:

```
# chkconfig nfsboot on
# chkconfig nfsserver on
```

When you restart the controller, these services will be started.

7.5.4 Turning off unneeded services

There are a number of services which are started in a SLES10 SP2 minimum system. They can be viewed through the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
haldaemon               0:off  1:off  2:off  3:on   4:off  5:on   6:off
network                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsboot                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver               0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

Archived

```
portmap                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
postfix                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
resmgr                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
slpd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
splash                  0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early            0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
suseRegister            0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
xinetd                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

In order to keep the controller as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services through the **chkconfig** command:

```
# chkconfig slpd off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig resmgr off
# chkconfig suseRegister off
```

You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
haldaemon               0:off  1:off  2:off  3:on   4:off  5:on   6:off
network                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsboot                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver               0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
portmap                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
xinetd                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

7.5.5 Applying service if necessary - online update

You may want to apply service through YaST Online Update. Internet access was not available during the time of writing, so step-by-step details are not available.

If you have access to the Internet, or an online update source, invoke **yast** => **Software** => **Online update**.

7.5.6 Installing the cmsfs package

The **clone.sh** script requires the **cmsfs** package, written by Rick Troth, in order read the CMS files. The cmsfs package is part of SLES10 SP2 distribution. Install it through the **yast -i** command:

```
# yast -i cmsfs
```

You should see some screens flash by as the cmsfs RPM is installed:

```
# rpm -qa | grep cmsfs
cmsfs-1.1.8-3.2
```

To test that the cmsfs package is properly installed, see if you can read the LINUX00 PARM-S10 parameter file. First you need to bring the 191 disk online through the **chccwdev -e** command. Then view the DASD to verify that the system knows about through the **lsdasd** command:

```
# chccwdev -e 191
Setting device 0.0.0191 online
Done
# lsdasd
0.0.0100(ECKD) at (94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(ECKD) at (94:  4) is dasdb : active at blocksize 4096, 600840 blocks, 2122 MB
0.0.0102(ECKD) at (94:  8) is dasdc : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0103(ECKD) at (94:12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at (94:16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0300(FBA ) at (94:20) is dasdf : active at blocksize 512, 524288 blocks, 256 MB
0.0.0301(FBA ) at (94:24) is dasdg : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0191(ECKD) at (94:28) is dasdh : active at blocksize 4096, 54000 blocks, 210 MB
```

Test some of the cmsfs utilities. The **cmsfslst** command lists files on a minidisk:

```
# cmsfslst -d /dev/dasdh
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
         DIRECTOR PO F         64         22          1  4/11/2008 12:17:53
         ALLOCMAP PO F       4096          2          2  4/11/2008 12:17:53
CHPW52   XEDIT    B1 V         70        180          3  5/15/2008 12:47:06
CPFORMAT EXEC     B1 V         79        231          3  5/15/2008 12:47:07
LABEL540 EXEC     D1 V         76        110          2  5/15/2008 12:56:44
LABEL540 XEDIT    D1 V         71         41          1  5/15/2008 12:57:42
LBL540-9 EXEC     D1 V         75         99          1  5/15/2008 17:11:13
LBL540-9 XEDIT    D1 V         71         43          1  5/19/2008 16:56:36
LINUX00  PARM-S10 D1 F         80          9          1  5/19/2008 15:24:57
PROFILE  EXEC     D1 V         63         17          1  5/19/2008 15:36:43
SLES10S2 EXEC     D1 V         69          9          1  5/16/2008 16:24:52
SLES10S2 INITRD   B1 F         80     106644       2083  5/16/2008 15:18:40
SLES10S2 KERNEL   B1 F         80      89413       1229  5/16/2008 15:18:27
SLES10S2 PARM-S10 B1 V         65          9          1  5/15/2008 12:47:14
SWAPGEN  EXEC     B1 V         72        358          5  5/15/2008 12:47:17
```

The **cmsfscat** command types the contents of a file:

```
# cmsfscat -d /dev/dasdh -a linux00.parm-s10
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.5.30 Hostname=virtc530.itso.ibm.com
Gateway=9.12.4.1 Netmask=255.255.252.0
Broadcast=9.12.4.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7 Portname=dontcare
Install=nfs://9.12.4.185:/install/sles10sp2/SLES-10-SP2-DVD-s390x-RC3-DVD1.iso
UseVNC=1  VNCPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

7.5.7 Enabling the vmcp and cmm modules

The **vmcp** module/command allows z/VM CP commands to be issued from Linux. It is critical to the functioning of the **clone.sh** script.

The **cmm** module allows Linux to do cooperative memory management (CMM1) with z/VM. This feature must be enabled on *both* the Linux and z/VM sides in order for it to function (hence the name “cooperative”). To enable z/VM for CMM1, see 11.7, “Enabling VMRM to work with CMM” on page 198.

To configure the **vmcp** and **cmm** modules to be loaded at boot time, edit the file /etc/sysconfig/kernel and add the module names to the variable MODULES_LOADED_ON_BOOT (around line 30):

```
# cd /etc/sysconfig
# vi kernel   // add vmcp to MODULES_LOADED_ON_BOOT
...
## Type:                string
## ServiceRestart:      boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
#
MODULES_LOADED_ON_BOOT="vmcp cmm"
...
```

Save the file and you should be able to issue CP commands through the **vmcp** after your system is rebooted.

7.5.8 Setting system to halt on SIGNAL SHUTDOWN

The Ctrl-Alt-Del key sequence is simulated by z/VM when it issues a **SIGNAL SHUTDOWN** command. Rather than rebooting, you want your system to halt (shutdown). Make a backup copy of the original, then change this setting by changing **shutdown -r** to **shutdown -h** in the /etc/inittab file:

```
# cd /etc
# cp inittab inittab.orig
# vi inittab    // change shutdown -r to shutdown -h
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

This change will be picked up when the system is rebooted.

7.5.9 Modifying zipl.conf

When a SLES10 SP2 system boots, there are two ways it can determine which disks to bring online.

- The default method is that the device addresses are built into the initial RAMdisk.
- The disks to bring online can also be written to the zipl configuration file, /etc/zipl.conf, after which the **zipl** command must be run.

We recommend that you use the second method so it will be easier to determine which minidisks are designed to be part of the system. It is much easier to read a file (/etc/zipl.conf) than to find where the disks are specified in the initial RAMdisk.

Three changes are recommended to the default zipl.conf file:

1. The dasd= parameter can be added to parameter line. When specifying it you must maintain the ordering of the existing minidisks (/dev/dasda = 100, /dev/dasdb = 101 and /dev/dasdc = 102, and so on).
2. The parameter vmpoff=LOGOFF has the effect of logging the z/VM user ID off when Linux is shut down. This can be convenient for shutting the z/VM system down more efficiently and also for getting a refreshed 3270 emulator session.

3. The `timeout` value is modified to 3 seconds. When SLES 10 SP2 boots, the default time that is allowed to enter a menu command is 10 seconds. Because the default value is most commonly used, this 10 seconds delays the starting of Linux. It is backed up to 3 seconds.

Make a backup copy of the file and add the strings **dasd=100-104,300-301** and **vmpoff=LOGOFF** to the parameter line in the [ipl] section and set **timeout=3** in the *menu* section:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf   // add the dasd= string
# Modified by YaST2. Last modification on Wed Apr  9 19:02:36 UTC 2008
[defaultboot]
defaultmenu = menu

:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 3
    1 = ipl
    2 = Failsafe

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
    image = /boot/image
    target = /boot/zipl
    ramdisk = /boot/initrd,0x1000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-104,300-301
vmpoff=LOGOFF TERM=dumb"
###Don't change this comment - YaST2 identifier: Original name: failsafe###
[Failsafe]
    image = /boot/image-2.6.16.60-0.3-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.16.60-0.3-default,0x1000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1   TERM=dumb 3"
```

Now, run the **zipl** command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'ipl' (default)
Adding #2: IPL section 'failsafe'
Preparing boot device: dasda (0100).
Done.
```

These changes will be utilized the next time Linux is rebooted.

7.5.10 Rebooting the system

You should now reboot the system to test the changes:

```
# reboot
```

After your system comes back, start a new SSH session to the controller.

7.5.11 Verifying the changes

You are now done customizing the controller Linux image. SSH back into the controller and check a few settings. Test the **vmcp** command with a CP command such as **QUERY NAMES**:

```
# vmcp q n
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
LINUXOO  - DSC
VSM     - TCPIP
```

Confirm that both of your swap spaces are operational:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdf1                             partition       259956  0
/dev/dasdg1                             partition       519924  0
```

Verify the NFS server is running:

```
# rcnfsserver status
Checking for kernel based NFS server:                                   running
```

Test mounting the SLES 10 SP2 install directory:

```
# ls /mnt
# mount localhost:/nfs/sles10sp2 /mnt
# ls /mnt
SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
```

This shows that you can mount the directory with the SLES 10 SP2 install DVD. Now unmount it:

```
# umount /mnt
```

At this point, you have installed and configured a SLES 10 SP2 Linux system onto the controller. The next step is to install and configure the golden image.

Archived

Archived

8

Installing SLES10 SP2 on the golden image

Note: Chapters Chapter 4, “Installing and configuring z/VM” on page 27, 5, 6, and 7 must be completed before you proceed with this chapter.

This chapter describes how to install SLES 10 SP2 onto the user ID SLES10S2, which is referred to as the *golden image*. The golden image is the copy of Linux that will be cloned. Normally the system is shut down and the user ID logged off because it is not recommended to clone a running Linux system.

In this example, the golden image is given two 3390-3s at minidisk addresses 100 and 101. This allows for about 4.5 GB of disk space for each Linux system. If you want to increase that size, we recommend that larger volumes such as 3390-9s be used, but that minidisks at addresses 100 and 101 still be used. This will allow the **clone.sh** script to continue to function.

To install and configure the golden image, perform the following steps:

- Creating the SLES10S2 user ID
- Creating the SLES10S2 parameter file
- Updating AUTOLOG1 PROFILE EXEC
- Installing the golden image
- Configuring the golden image

© Copyright IBM Corp. 2008. All rights reserved.

8.1 Creating the SLES10S2 user ID

The golden image has a default memory size of 512 MB, and is just given class G privilege. It is given the following minidisks:

- 100 - Half of the disk space for the golden image.
- 101 - The other half of the disk space.

To accomplish this, perform the following steps:

1. Logon to MAINT.
2. Edit the USER DIRECT file, add six new lines at the bottom of the file, and create the following user directory entry. Set the 3390 disk labels to those appropriate for your system. In this example, MMD955 and MMD956 are used:

```
==> x user direct c
====> bot
====> a 6
*
USER SLES10S2 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3338 <MMD955> MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 0001 3338 <MMD956> MR LNX4VM LNX4VM LNX4VM
```

3. Go back to the top of the file and search for string USER $ALLOC$. Add cylinder 0 of each of the two new volumes to this dummy user ID so they do not show up as gaps.

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 <MVD850> R
 MDISK A02 3390 000 001 <MVD853> R
 MDISK A03 3390 000 001 <MVD854> R
 MDISK A04 3390 000 001 <MMD857> R
 MDISK A05 3390 000 001 <MMD856> R
 MDISK A06 3390 000 001 <MMD952> R
 MDISK A07 3390 000 001 <MMD953> R
 MDISK A08 3390 000 001 <MMD954> R
 MDISK A09 3390 000 001 <MMD955> R
 MDISK A0A 3390 000 001 <MMD956> R
...
====> file
```

4. Run **DISKMAP** to check for overlaps and gaps. You should only see the single 501 cylinder gap.

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
====> pre off
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  343  line(s) not displayed --------------------
====> quit
```

5. When the disk layout is correct, run **DIRECTXA** to bring the changes online:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
```

```
HCPDIR494I User directory occupies 44 disk pages
```

You have now defined the user ID that will contain the Linux golden image.

8.2 Creating the SLES10S2 parameter file

A SLES 10 SP2 parameter file will be needed for this new user ID. You will need to change the IP address (`HostIP` variable) and the host name (`Hostname` variable). In this example, those are 9.12.5.29 and virtc529.

Also, the controller is used to provide the SLES 10 SP2 install ISO image by setting the `Install` variable.

Perform the following steps:

- Logon to `LNXMAINT`.
- Copy the `LINUX00` parameter file to one with a file name of `SLES10S2` on the `LNXMAINT 192` (D) disk:

```
==> copy linux00 parm-s10 d sles10s2 = =
```

- Edit the new file and set the networking values correctly:

```
==> x sles10s2 parm-s10
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=<9.12.5.29> Hostname=<virtc529>.itso.ibm.com
Gateway=9.12.4.1 Netmask=255.255.252.0
Broadcast=9.12.4.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7 Portname=dontcare
Install=nfs://<9.12.5.30>/nfs/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
UseVNC=1  VNCPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

When `SLES10S2` is logged onto, this new parameter file will be accessible on its A (191) disk.

8.3 Updating AUTOLOG1 PROFILE EXEC

The new Linux ID you defined needs access to the VSWITCH. A **`SET VSWITCH`** command with the **`GRANT`** parameter can be added to the `AUTOLOG1` **`PROFILE EXEC`** to do this.

Perform the following steps

- Logon to `AUTOLOG1`.
- Issue the **`ACCESS (NOPROF`** command so the `PROFILE EXEC` is not run:

```
==> acc (noprof
```

- Add a line to grant the `SLES10S2` user ID access to the VSWITCH:

```
==> x profile exec
...
/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant linux00'
'cp set vswitch vsw1 grant sles10s2'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog linux00'
```

```
'cp logoff'                          /* logoff when done */
====> file
```

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

```
==> set vswitch vsw1 grant sles10s2
Command complete
```

8.4 Installing the golden image

You should now be ready to begin the install onto the golden image. Linux will be installed onto the 100-101 minidisks and 300-301 virtual disks. Most Linux user IDs described in this book will have two read/write minidisks and two VDISKS. Both swap disks are VDISKs, which means they are in z/VM virtual memory and thus provide fast access. Disk 300 is 256 MB and will act as a primary swap space. Only after it is full will disk 301 (512 MB) be used.

To install the golden image, perform the following steps:

1. Logon to **SLES10S2**. When you logon, you should see message indicating that a virtual NIC has been created at address 0600 and that VDISKs 300 and 301 have been created:

```
LOGON SLES10S2
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 08:33:24 EDT TUESDAY 05/20/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
```

2. You are prompted to IPL Linux, but since you have not installed Linux yet, answer n:

```
Do you want to IPL Linux from minidisk 100? y/n
==> n
```

3. Before you install Linux, it is good practice to verify the resources. Verify there are minidisks at virtual addresses 100 and 101 and virtual disks at addresses 300 and 301 through the **QUERY** command:

```
==> q 100-101
00: DASD  3390 MMD955 R/W       3338 CYL ON DASD  D955 SUBCHANNEL = 0000
00: DASD 0101 3390 MMD956 R/W       3338 CYL ON DASD  D956 SUBCHANNEL = 0001
==> q 300-301
00: DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
00: DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
```

4. Verify that you have a virtual OSA at addresses 600-602 with the **QUERY VIRTUAL OSA** command:

```
==> q v osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
00:      0600 DEVTYPE OSA         CHPID 20 OSD
00:      0600 MAC 02-00-01-00-00-03 CURRENT
00:      0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
00: OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0003
```

```
00:      0601 DEVTYPE OSA         CHPID 20 OSD
00:      0601 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
00: OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 0004
00:      0602 DEVTYPE OSA         CHPID 20 OSD
00:      0602 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
```

5. Use the **QUERY VIRTUAL STORAGE** command to show that you have a 512 MB machine:

```
==> q v stor
00: STORAGE = 512M
```

This shows that you have the resources necessary to install SLES 10 SP2.

8.4.1 Begin the SLES10 SP2 installation

Follow these steps to begin the installation of SLES10S2.

1. Run the **SLES10S2 EXEC**. You should see many screens of questions and answers scrolling by. If you had used the default parameter file shipped with SLES10S2, you would have had to answer all the networking questions manually.

   With the proper parameters set in the file SLES10S2 PARM-S10, the install process should proceed to where you access the install program through a VNC client:

```
==> sles10S2
sles10s2
00:      NO FILES PURGED
00: RDR FILE 0001 SENT FROM SLES10S2 PUN WAS 0001 RECS 089K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0002 SENT FROM SLES10S2 PUN WAS 0002 RECS 0009 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0003 SENT FROM SLES10S2 PUN WAS 0003 RECS 107K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Linux version 2.6.16.60-0.17-default (geeko@buildhost) (gcc version 4.1.2 200701
15 (SUSE Linux)) #1 SMP Tue Apr 22 07:28:35 UTC 2008
We are running under VM (64 bit mode)
Detected 2 CPU's
Boot cpu address  0
Built 1 zonelists
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dum
b                    HostIP=9.12.5.29 Hostname=virtc529.itso.ibm.com
                     Gateway=9.12.4.1 Netmask=255.255.252.0
                     Broadcast=9.12.4.255 Layer2=0
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
0.0602               Nameserver=9.12.6.7 Portname=dontcare
                     Install=nfs://9.12.5.30/nfs/sles10sp2/SLES-10-SP2-DVD-s390x
-GMC-DVD1.iso        UseVNC=1  VNCPassword=lnx4vm
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
...
```

The install system or *starter system* should continue to boot. You should see the message:

```
***
***           You can connect to 9.12.5.29, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.12.5.29:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)
```

```
*** Starting YaST2 ***
Xlib:  extension "XInputExtension" missing on display ":0.0".
```

2. You can open a Java-enabled browser to access YaST2 at the specified URL. The logon prompt in Figure 8-1 shows VNC access through a Java-enabled browser.

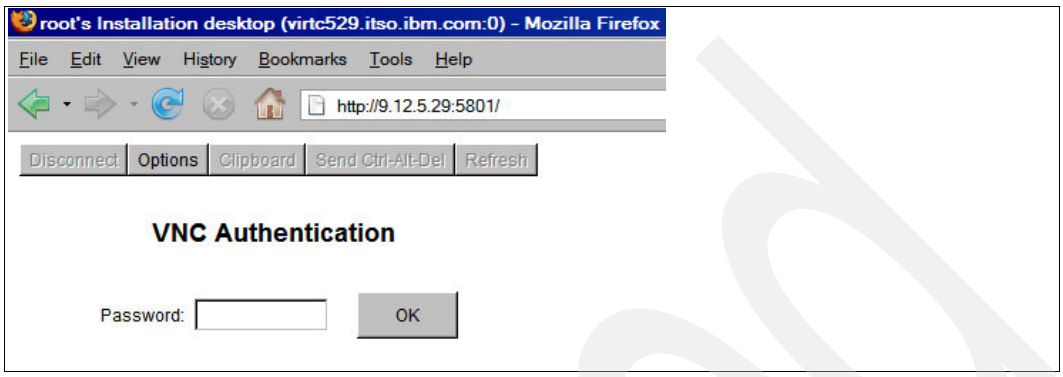


Figure 8-1 VNC viewer through a Java-enabled browser

In addition to a browser, you can also use a standalone VNC viewer if desired. If this is the case, the server to connect to would be **`9.12.5.29:1`** in this example, as shown in Figure 8-2. Enter the password specified in the parameter file (`lnx4vm`, in this example).

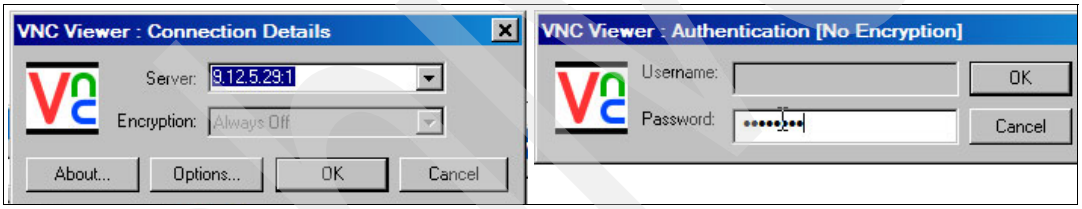


Figure 8-2 Using the VNC viewer

3. You could disconnect from the 3270 session, but messages to the console will be lost. We recommend that you stay connected, so you will have to clear the window periodically (or the install process may be delayed waiting for the window to clear itself).

Now the installation process should begin.

8.4.2 Beginning YaST installation

The installation program that is running is **`yast2`**. Perform the following steps:

1. Choose your language, (**English US**, in this example) and click **Next**.
2. The SLES10 SP2 license agreement window should appear. Choose **Yes, I agree to the License Agreement** and click **Next**.
3. The Disk Activation window should appears. Choose **Configure DASD Disks**.
4. The DASD Disk Management window should appear; you will see all the DASD available to `SLES10S2`.
   a. Highlight each of the mindisks and VDISKs, 100, 101, 300 and 301 and click **Select or Deselect**.
   b. You should see a **Yes** appear next to them in the Selected column. Activate them by clicking **Perform Action** -> **Activate**, as shown in the left side of Figure 8-3 on page 123.

c. Disks 100 and 101 must be formatted so that Linux can use them. Deselect disks 300 and 301 through the **Select or Deselect** button, so that 100 and 101 remain selected. Now click **Perform Action** -> **Format** as shown on the right side of the figure.

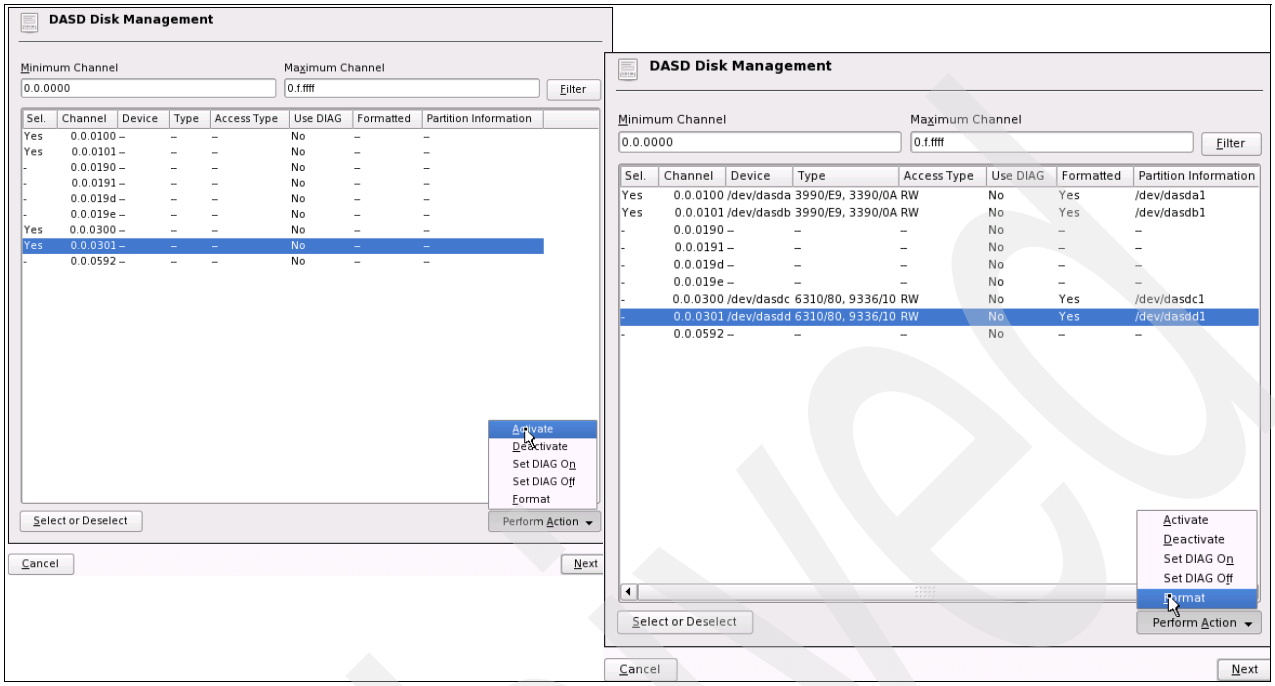


Figure 8-3 DASD that are available to SLES10S2

5. You should see a window asking for 2 Parallel Formatted Disks. Accept the default of writing a volume label and click **OK**.
6. Click **Yes** to the question `Really format the following disks?`
7. A progress indicator window should appear displaying the progress of the parallel formats. This step can take 1 to 10 minutes, depending on a number of factors.
8. When the formatting is complete, click **Next** in the DASD Disk Management window.
9. In the Disk Activation window click **Next** again.
10. In the Installation Mode window, accept the default of **New installation** and Click **Next**.
11. The Clock and Time Zone window will appear. Choose your region and time zone and click **Next**.
12. This will bring you to the Installation Settings window. Click **Partitioning** in the Overview tab. The Expert Partitioner window will appear as shown in Figure 8-4 on page 124.

Redbooks

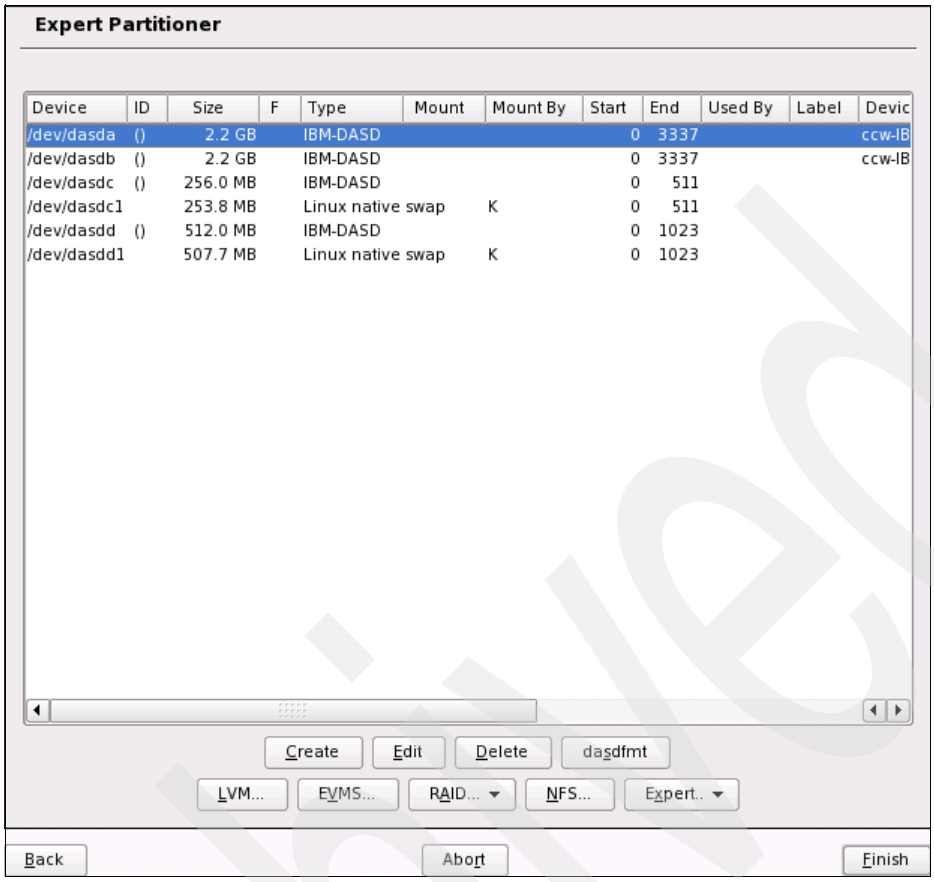


Figure 8-4 Disk partitioner - before customization

13. In this example, two partitions are made on `/dev/dasda` and one partition is made on `/dev/dasdb`. Click **Create** and you should see a new window named Disk to Partition.
14. Select /dev/dasda and click OK. You should see a new window Create a Primary Partition on /dev/dasda, as shown on the left side of Figure 8-5 on page 125. Perform the following steps:
    a. Click the **Format** radio button.
    b. Choose **Ext3** as the file system.
    c. Choose a size of **384 MB** or the size you have decided upon.
    d. Click the **Fstab Options** button and select **Device path** as shown on the right side of Figure 8-5 on page 125. This is necessary for cloning as the default of Device ID will tie a specific disk ID to the `/etc/fstab` file. Click **OK**.
    e. Accept the default mount point of **/**.
    f. Click **OK**. This should create the partition `/dev/dasda1`.

Archived

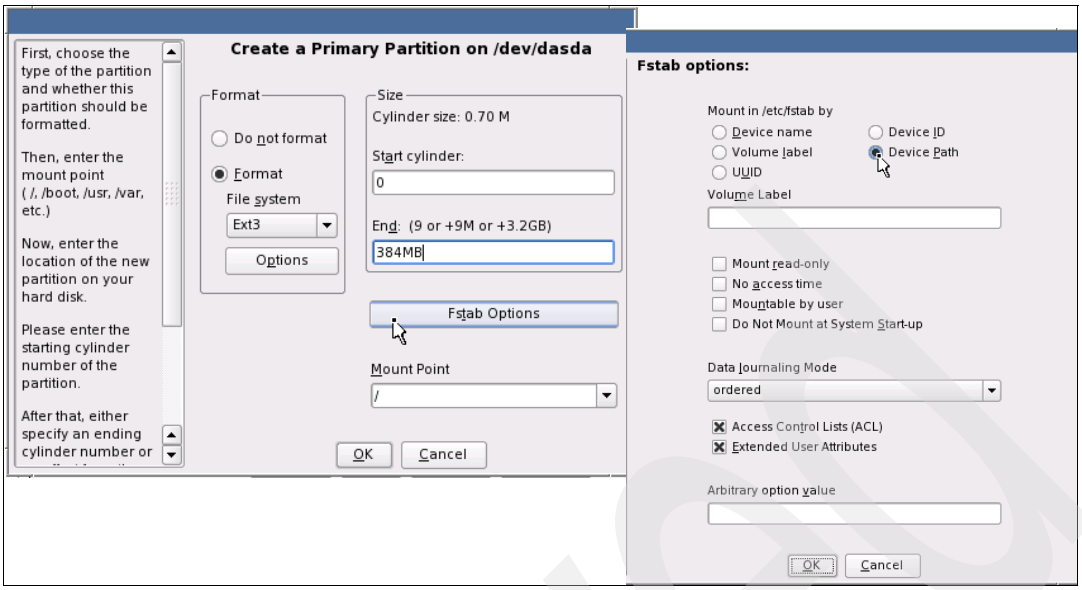


Figure 8-5 Creating partition for root file system

15. To create a second partition, click **Create** then choose `/dev/dasda` again. You should see a new window Create a Primary Partition on /dev/dasda as shown in Figure 8-6. This time perform the following steps:
    a. Click the **Do not format** radio button.
    b. Accept the default size of all the remaining space
    c. Select an empty Mount Point.
    d. Click **OK**. This should create an empty partition `/dev/dasda2`.

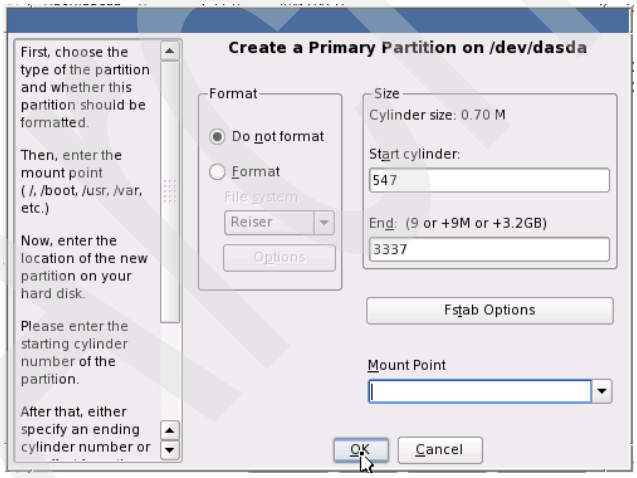


Figure 8-6 Creating the empty partition /dev/dasda2

16. Create an empty partition `/dev/dasdb1` in the same way as you created `/dev/dasda2` (**Create** => `/dev/dasdb` => **Do not format**, empty Mount Point). Accept the default size of all the space and click **OK**.
17. Now `/dev/dasda2` and `/dev/dasdb1` can be used to create a volume group. Click **LVM** as shown on the left side of Figure 8-7 on page 126.

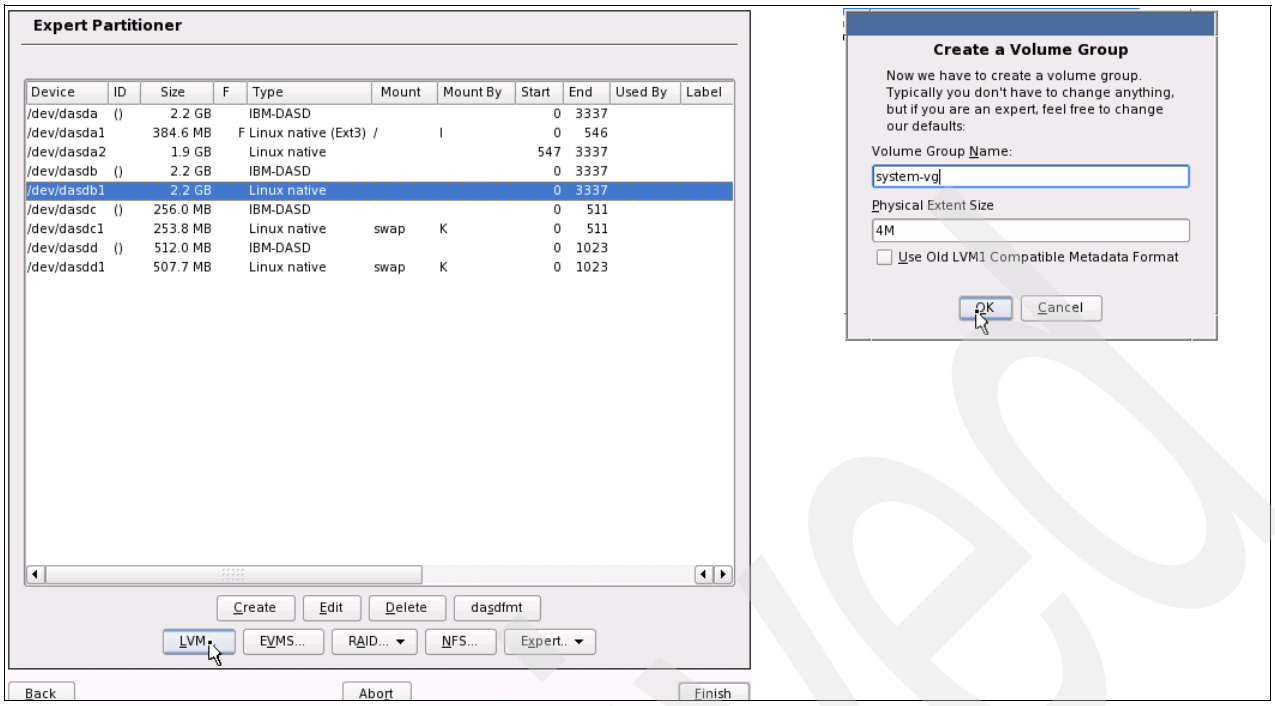


Figure 8-7 Creating a volume group and logical volumes

18. At the Create a Volume Group window as shown on the right side of Figure 8-7, set the volume group name to **system-vg** and click **OK**.

19. The Logical Volume Manager: Physical Volume Setup window should appear as shown on the left side of Figure 8-8. Click **Add Volume** and `/dev/dasda2` will be added. Also add `/dev/dasdb1`. Click **Next**.

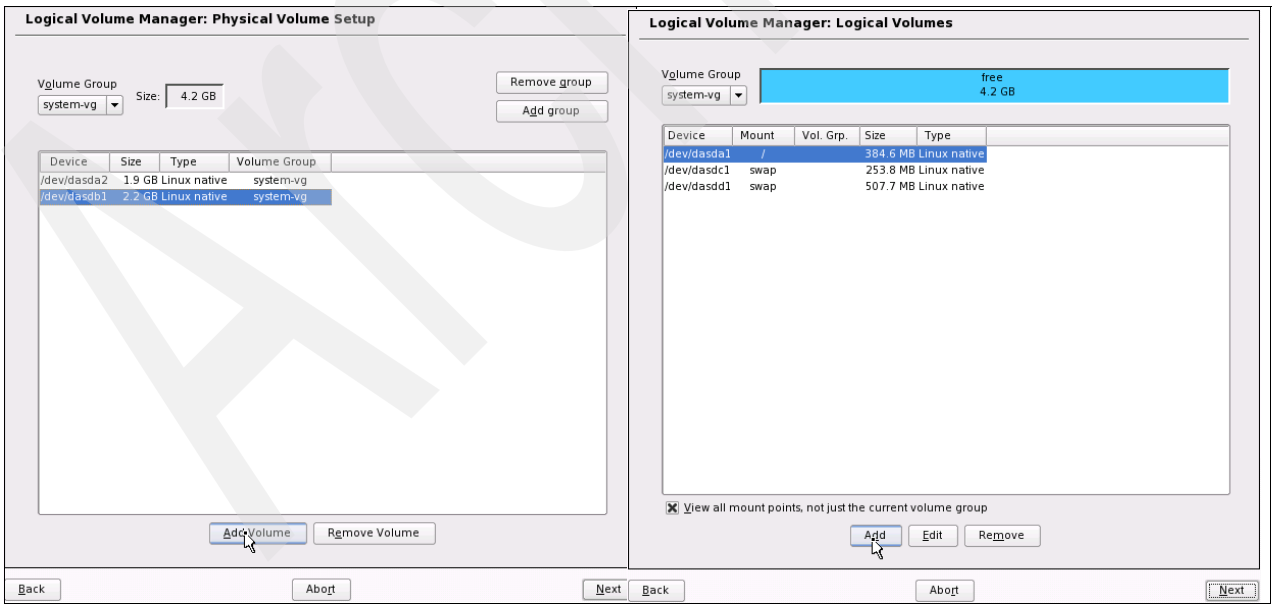


Figure 8-8 Setting up phsical volumes

20. The Logical Volume Manager: Logical Volumes window should appear as shown on the right side of Figure 8-8. Click **Add** to create a logical volume.

21. The left side of Figure 8-9 shows adding a Logical volume name of **usr-lv** with a size of **2 GB** and a File system of type **ext3**. Click **OK** when complete.

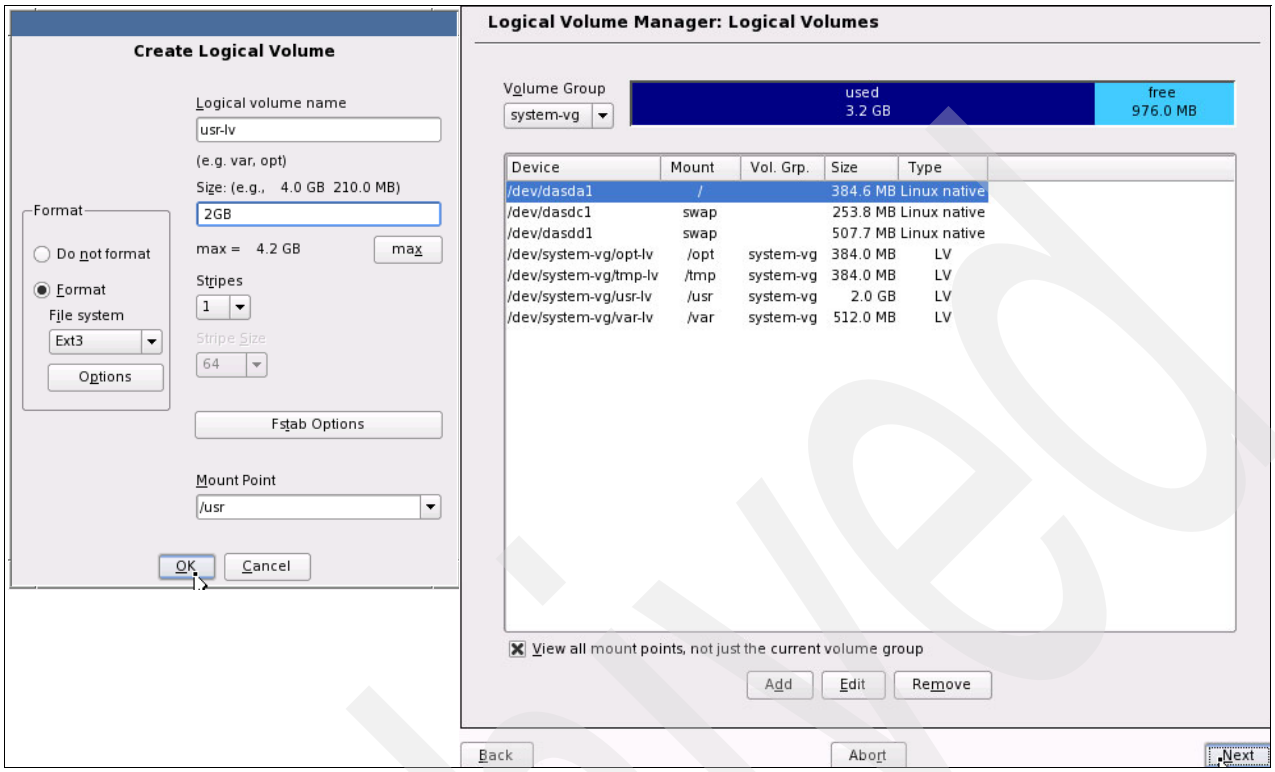


Figure 8-9 Adding logical volumes

22. Create three more logical volumes, **var-lv**, **opt-lv**, and **tmp-lv**, similarly using Table 8-1.

Table 8-1 Logical volumes added to system-vg volume group

Mount point	Logical volume name	Size
/usr/	usr-lv	2 GB
/var/	var-lv	512 MB
/opt/	opt-lv	384 MB
/tmp/	tmp-lv	384 MB

You may choose other sizes. See 2.3, “Disk planning” on page 10 for more discussion about this topic.

When all logical volumes are created, you should see a summary as shown on the right side of Figure 8-9. In this example, 3.2 GB are used and about 1 GB is available for growth. Click **Next**.

23. You do not have to format /dev/dasdc1 (vaddr 300) and /dev/dasdd1 (vaddr 301) because they were properly formatted as a Linux swap space by the **SWAPGEN EXEC**, and thus should be recognized as a swap space.

24. Double-click the swap space **/dev/dasdc1**. The Edit Existing Partition /dev/dasdc1 window will appear.

25. Click the **Fstab options** button as you did previously. The Fstab options window will appear.

26. Select the **Device Path** radio button in Mount in /etc/fstab by radio group. Click **OK**.
27. In the Edit Existing Partition /dev/dasdc1 window, click **OK**.
28. Repeat the previous steps for the second swap space, **/dev/dasdd1**.
29. Back in the Expert Partitioner window, you should see a window similar to Figure 8-10. Click **Finish**.

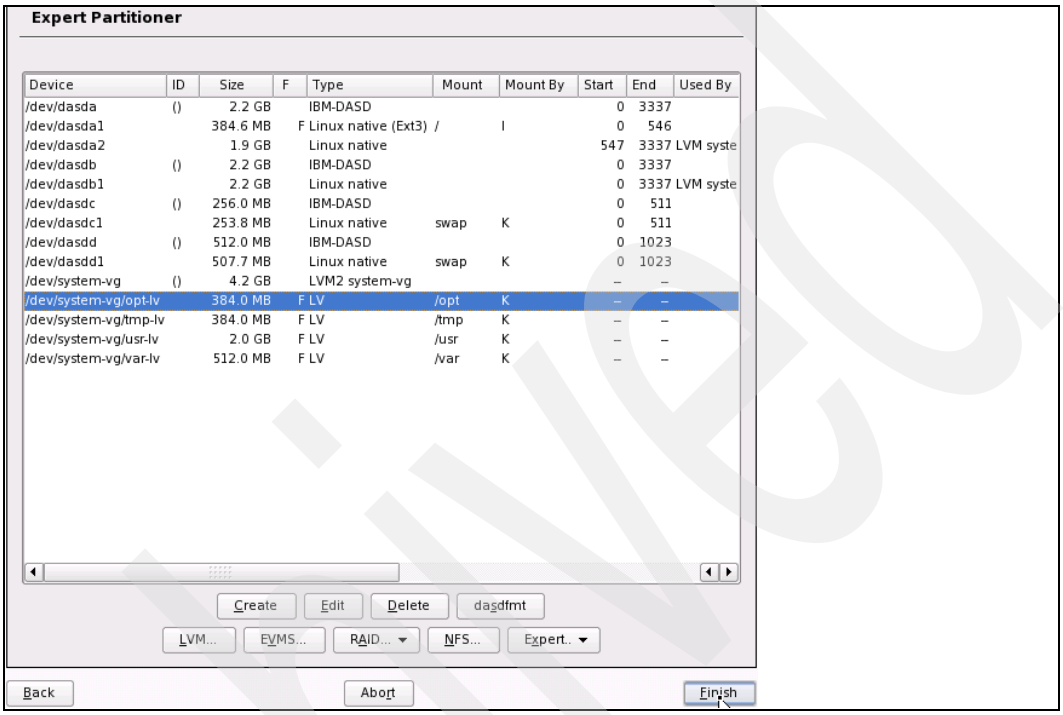


Figure 8-10 Disk partitioner - after customization

30. In the Installation Settings window, select **Software**.
31. The Software Selection window opens. Leave only **Server Base System** selected. Uncheck all other setting as showed in Figure 8-11 on page 129**,** click **Accept**.
32. You may be prompted to accept a license agreement for agfa fonts.

Archived

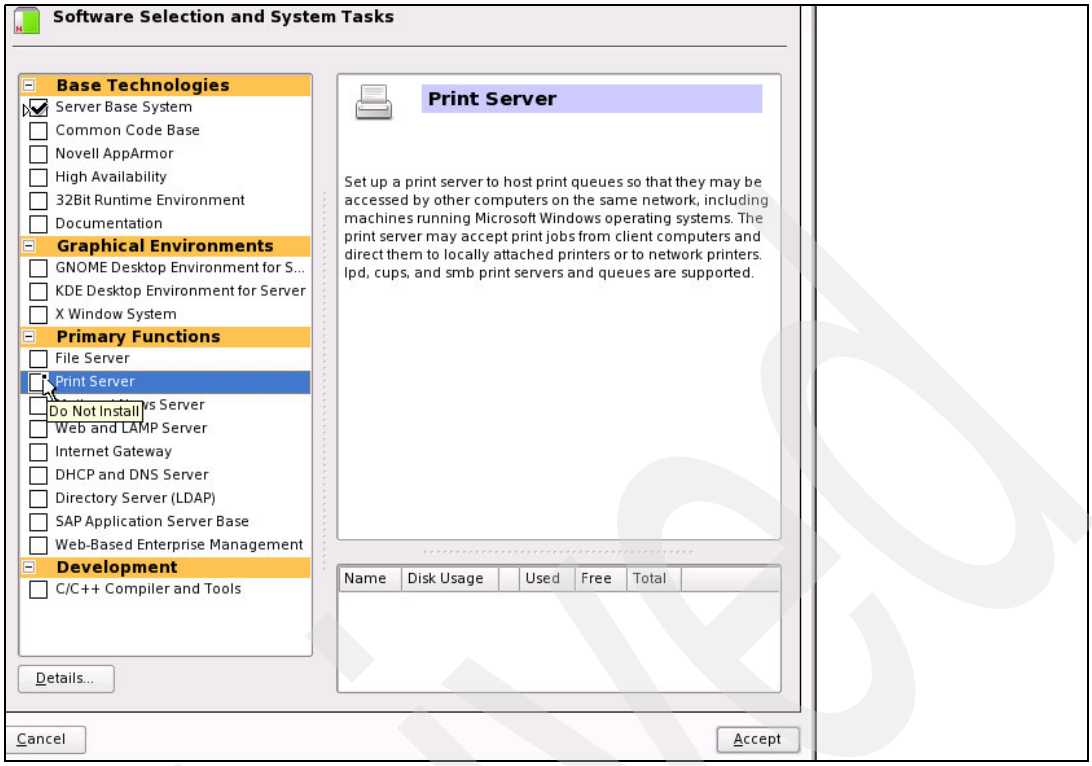


Figure 8-11 Software selection

33. In the Languages window, select your primary and optionally secondary languages and click **Accept**.
34. You are now ready to begin copying the RPMs onto your root file system. In the Installation Settings window, check the settings and click **Accept**.
35. In the Confirm Installation window, click **Install**.

A minimal SLES10 SP2 system will be installed onto DASD. This should take about 5 to 20 minutes depending on a number of factors.

8.4.3 Booting your new Linux system from disk

After the first part of installation completes, your Linux system will shut down and you will need to return to your z/VM 3270 session. **IPL** the newly installed system from disk to continue installation. Issue the following command and your new kernel should boot from disk:

```
...
Power down.
01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from
 CPU 00.
00: HCPGSP2630I The virtual machine is placed in CP mode due to a SIGP stop and
store status from CPU 00.
```

Enter **IPL 100** and Linux will be IPLed.

```
==> ipl 100
00: zIPL v1.6.3 interactive boot menu
00:
00:  0. default (ipl)
```

```
00:
00:  1. ipl
00:  2. Failsafe
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 10 seconds):
```

At this prompt, you can wait 10 seconds for the default IPL to proceed, or you can try to type `#cp vi vmsg 0` if you can type quickly enough.

```
00: Booting default (ipl)...
Linux version 2.6.16.60-0.3-default (geeko@buildhost) (gcc version 4.1.2 20070115
(prerelease) (SUSE Linux)) #1 SMP Thu Jan 31 23:35:12 UTC 2008
We are running under VM (64 bit mode)
...
```

You will have to clear the window a number of times. The install program will bring up a VNC server again to complete the installation.

```
...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.12.5.29, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.12.5.29:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***
...
```

8.4.4 Completing YaST2 installation

Go back to the same VNC client used for the first part of installation. If it was a browser, either click **Login Again** or click the browser's **refresh** button until another VNC login window appears.

If you are using a VNC client, then open that application again.

1. Log in using the same VNC password (`lnx4vm`, in this example) as used to first bring up YaST2.
2. Now you will get a window for setting the root password. Enter your desired root password twice and click **Next**. It is important to remember this password.
3. In the Hostname and Domain Name window, enter the required DNS values. In this example, `virtc529` is the host name and `itso.ibm.com` is the domain name. Uncheck the **Change Hostname via DHCP** check box. Click **Next**.
4. In the Network Configuration window, you will see `Firewall is enabled`. Click the word **enabled** to change it to **disabled**, as shown in Figure 8-12 on page 131. All other values should be correct, so click **Next**.

Archived

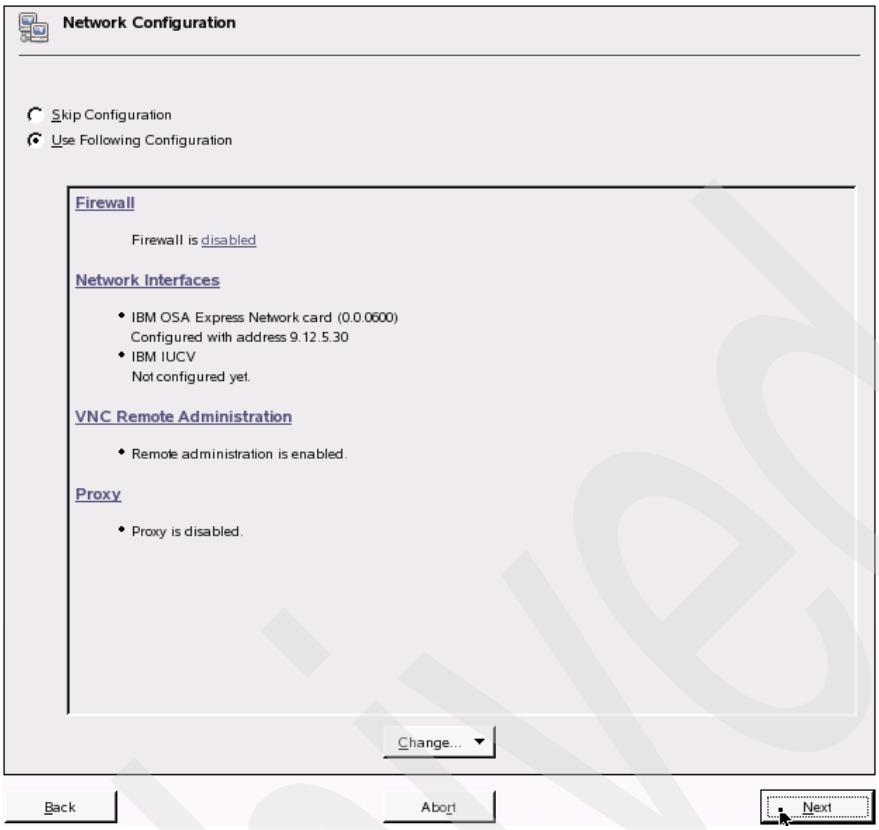


Figure 8-12 Disabling firewall in Network Configuration window

5. In the Test Internet Connection window, select **No, skip this test** and click **Next**.
6. In the Installation Settings window, accept the default and click **Next**. A default CA and certificate will be created.
7. In the User Authentication Method window, accept the default of **Local (/etc/passwd)** and click **Next**.
8. The next window should be Add a new local user. We recommend that you add at least one user so as to have a *non-root* ID on all cloned systems. In this example, the user `mikemac` was added. When you are done, click **Next**.
9. In the Writing the system configuration window, the **`SuSEconfig`** tool writes all your settings to disk.
10. The next window will be Release Notes. After reviewing the release notes, click **Next**.
11. In the Hardware Configuration window, choose the **Skip Configuration** radio button and click **Next**.
12. The last installation window is Installation Completed. **Uncheck** the check box Clone This System for Autoyast box and click **Finish.**

The VNC session should end. Return to the 3270 session; you may have to clear the window a few times. Then you should see a login prompt. At this point, you have completed the installation of Linux. You can disconnect from the 3270 session through the command:

```
==> #cp disc
```

From this point forward, we recommend that you access your Linux systems through SSH. If you have a Windows desktop, but do not have an SSH client configured, refer to 3.1, “PuTTY: a free SSH client for Windows” on page 20, for more information about that topic.

8.5 Configuring the golden image

Now you want to customize the golden image before cloning. The following high level steps are recommended, although you may add or omit some steps. However, omitting steps can negatively affect the operation of the **clone.sh** script.

- Configuring the VNC server
- Preparing for Online Update
- Removing unnecessary RPMs
- Turning off unneeded services
- Enabling the cmm module
- Configuring /etc/inittab
- Verifying the On Demand Timer Patch is set
- Configuring SSH keys
- Modifying zipl.conf
- Cleaning temporary files
- Rebooting the system and verifying changes

8.5.1 Configuring the VNC server

Start an SSH session as root to the golden image.

Often applications require a graphical environment. The `tightvnc` package is a Virtual Network Computing (VNC) server. It allows for a graphical environment to be set up easily through the **vncserver** command.

SLES10 SP2 configures a VNC connection that starts at the KDE prompt, by default. It is configured through the **xinetd** daemon. We recommend that you disable it. Edit the file **/etc/xinetd.d/vnc**. In sections `service vnc1` and `service vnchttpd1`, add the parameter **disable=yes**.

```
# cd /etc/xinetd.d
# vi vnc
   service vnc1
   {
           socket_type     = stream
           protocol        = tcp
           wait            = no
           user            = nobody
           server          = /usr/X11R6/bin/Xvnc
           server_args     = :42 -inetd -once -query localhost -geometry 1024x768 -depth 16
           type            = UNLISTED
           port            = 5901
           disable         = yes
   }
   ...
   service vnchttpd1
   {
           socket_type     = stream
           protocol        = tcp
           wait            = no
           user            = nobody
```

```
        server          = /usr/X11R6/bin/vnc_inetd_httpd
        server_args     = 1024 768 5901
        type            = UNLISTED
        port            = 5801
        disable         = yes

}
```

VNC is the only service that **xinetd** starts by default (this can be verified with the **chkconfig --list** command). So you can turn off the **xinetd** service with the **service xinetd stop** command for this session and the **chkconfig** command at boot time:

```
# service xinetd stop
Shutting down xinetd:                                      done
# chkconfig xinetd off
```

When you first start the VNC server, you are prompted to set a password. After it is set, this will be the password that you will need to connect to it from a VNC client:

```
# vncserver
You will require a password to access your desktops.

Password: lnx4vm
Verify: lnx4vm
Would you like to enter a view-only password (y/n)? n
xauth:  creating new authority file /root/.Xauthority

New 'X' desktop is virtc529:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
LLog file is /root/.vnc/virtc529:1.log
```

Stop the VNC server using the **-kill :1** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 6961
```

The openmotif package allows for the motif window manager (**mwm**) that is more usable than the Tiny Window Manager (**twm**) that VNC uses by default. We therefore recommend that you change to **mwm**. To add the openmotif RPM, you can use the **yast -i** command:

```
# yast -i openmotif
```

You should see **yast** curses screens go by as the software is added. Verify that the RPM was added:

```
# rpm -q openmotif
openmotif-2.2.4-21.12
```

Change the window manger from **twm** to **mwm** in the file /root/.vnc/xstartup:

```
# cd /root/.vnc/
# vi xstartup
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

You may want to remove the passwd file so the cloned system does not have the same password as you just entered. If so, use the command **rm passwd**. If you remove the

password file, when a system is cloned, the password will be prompted for the first time that the VNC server is initialized.

Verifying VNC server configuration

Start the VNC server again:

```
# vncserver

New 'X' desktop is virtc529:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtc529:1.log
```

Start a VNC client as shown on the left side of Figure 8-13. Enter the password and the resulting VNC session is shown on the right side of Figure 8-13.

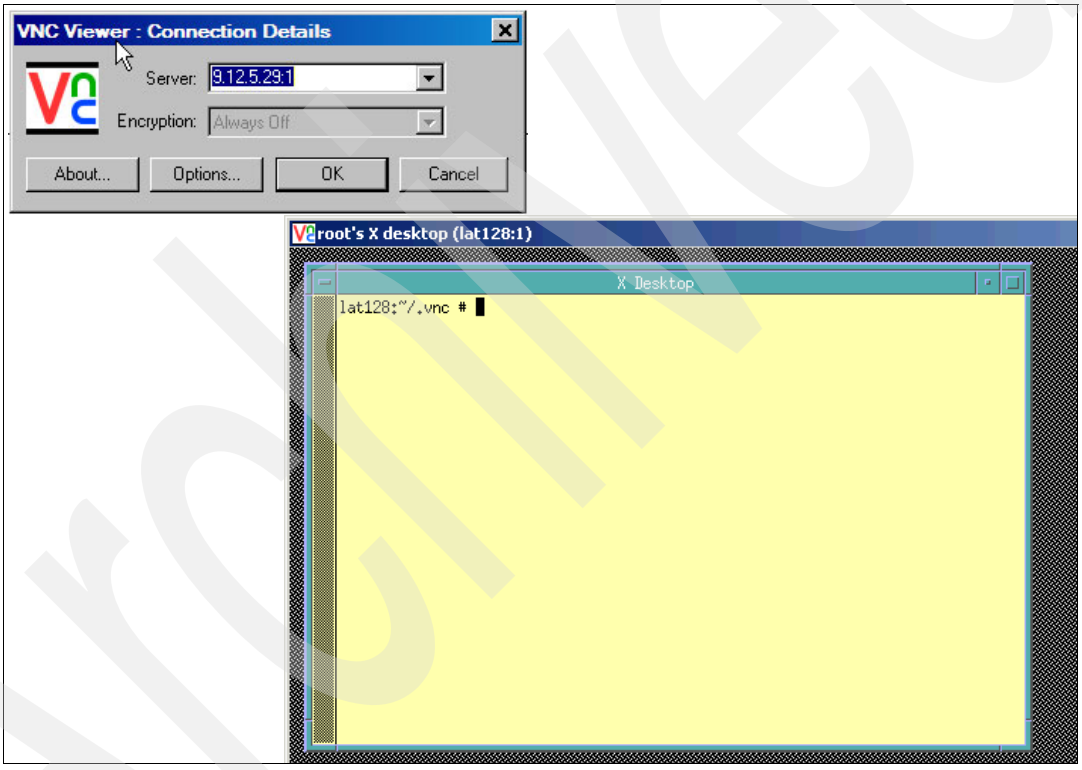


Figure 8-13 VNC session

Note that the VNC server will *not* be started across reboots. When you need a graphical environment, you will either have to start vncserver manually (recommended), or you will have to start it in the desired run level through a script such as `/etc/init.d/boot.local`.

Unless you need a graphical environment, you can again stop the VNC server using the **`-kill :1`** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 6961
```

Archived

8.5.2 Preparing for Online Update

This step is recommended if you have Internet access.

Note: The system used for the writing of this book did not have Internet access, so the following steps have not been fully tested for SLES 10 SP2.

Before you can use Online Update (recommended), online sources have to be configured. This is done through the Novell Customer Center Configuration. To configure the Customer Center, a Web browser is needed. For this reason, a VNC server session must be started.

A graphical environment is recommended for this step. Start a VNC viewer. In a terminal session, start YaST with command **yast2**.

Perform the following steps:

- Choose the **Novell Customer Center Configuration** selection.
- On next window, select **Configure Now** and click **Next**. In the Manual Intervention Required window, read the information and press **Continue**.
- Now mozilla-xulrunner (with many other packages) will be installed so that the browser can be run. Click **Continue** and wait until everything is installed and a browser is started. This will take a few minutes.
- In the Web Registration page enter your e-mail address twice and name of the system (optional). If you have activation code for SLES, you can enter it here. If you do not enter activation code, Novell will create temporary one that will expire in 15 days. Press the **Submit** button.
- On the next window, read the information and press **Continue**.
- Now the Contacting server... window will be active for a few minutes and online sources will be set in YaST. You can check it in the **Details...** window. Click **OK**.
- Click **Quit** to quit YaST.

Your system should now be configured to run Online Update.

8.5.3 Turning off unneeded services

A number of services are started in an SLES10 SP2 minimum system. They can be viewed through the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
haldaemon               0:off  1:off  2:off  3:on   4:off  5:on   6:off
network                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsboot                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
portmap                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
postfix                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
resmgr                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
slpd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
splash                  0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early            0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

```
suseRegister              0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

To keep the golden image as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services through the **chkconfig** command:

```
# chkconfig nfsboot off
# chkconfig slpd off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig resmgr off
# chkconfig suseRegister off
```

You may choose to leave these services on, or turn others off. You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
haldaemon                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
network                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                       0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
portmap                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

This shows the services that will run in the default run level of 5.

8.5.4 Enabling the cmm module

The **cmm** module allows Linux to do cooperative memory management (CMM1) with z/VM. This feature must be enabled on *both* the Linux and z/VM sides in order for it to function (thus the term *cooperative*). To enable z/VM for CMM1, see 11.7, “Enabling VMRM to work with CMM” on page 198. To enable it for Linux, perform the following steps.

Edit the file `/etc/sysconfig/kernel` and add **cmm** to the variable `MODULES_LOADED_ON_BOOT` (around line 30):

```
# cd /etc/sysconfig
# vi kernel
...
## Type:               string
## ServiceRestart:     boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
#
MODULES_LOADED_ON_BOOT="cmm"
...
```

Save the file, and the **cmm** module should be loaded after your system is rebooted.

8.5.5 Applying service - Online Update

If you have a SUSE Maintenance Web account, then you can use it to retrieve the latest patches for SLES10 SP2. Because many of these patches contain security and bug fixes, it is

recommended that you apply the patches for the golden image so that it is up-to-date. Subsequently all the servers you clone after the golden image will also be up-to-date.

Note: You must complete 8.5.2, “Preparing for Online Update” on page 135, before proceeding with this section.

The system used for writing this book did not have Internet access, so the following steps have not been fully tested.

There should be a VNC server running on the golden image. Connect with a VNC viewer. Perform the following tasks:

1. From a terminal session, start **yast2**.
2. In YaST, choose **Software** on the left and **Online Update** on the right.
3. You will be presented with list of downloadable patches. Patches to packages you have installed will be checked for download. You may choose here which packages to install and which not to install. We highly recommend downloading security patches.

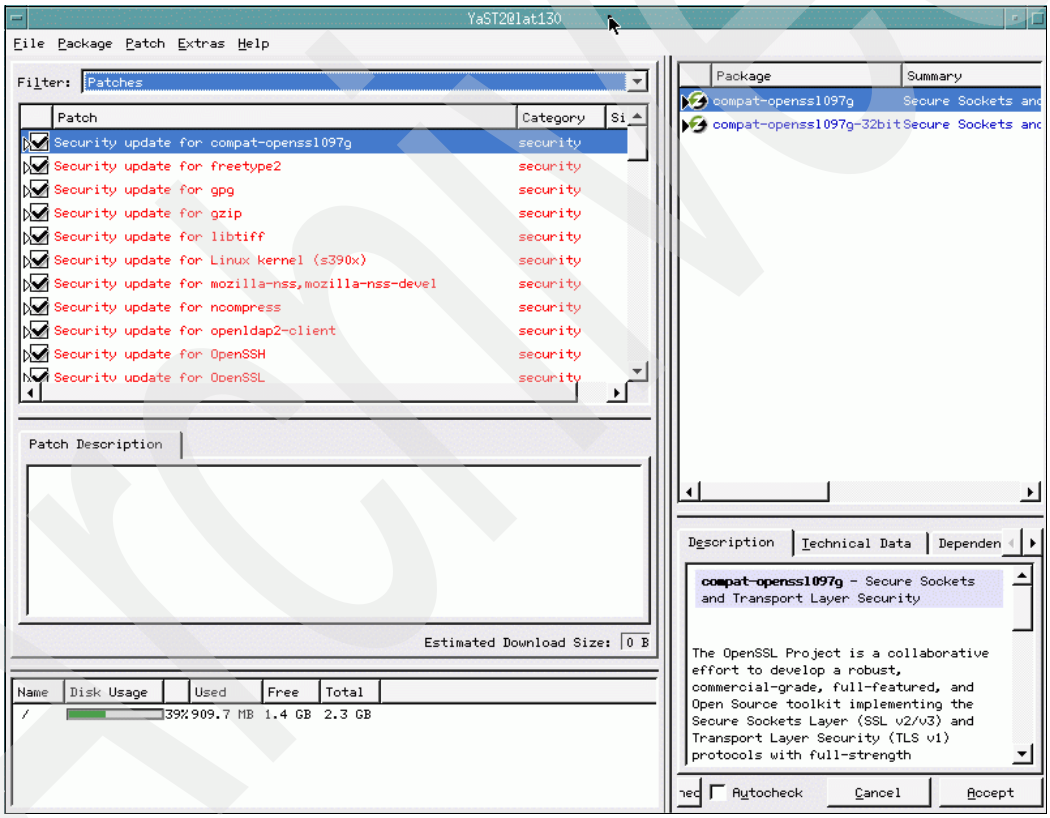


Figure 8-14 Available patches in Online Update

4. When done, click **Accept**. There may be other windows coming with warnings, so read them carefully. (For example, you may get a window informing you about a kernel security update that tells you to reboot the system after you finish.) Click **Accept**.
5. Patches will be downloaded and installed. It may take some time depending on the number of patches, their size, and your network speed.
6. Click **Finish** when install ends.
7. Click **Quit** to quit YaST.

When you finish here, do not reboot yet, because you will reboot shortly after some boot configuration changes are made.

8.5.6 Configuring /etc/inittab

By default, SLES 10 SP2 boots into run level 5, which is designed for a graphical environment. To keep the golden image as lean as possible, we recommend that this be reset to run level 3. This is set in the `/etc/inittab` file with the variable `initdefault`.

Linux reboots when a Ctrl-Alt-Del key sequence is trapped. This key sequence is simulated by z/VM when it issues a **`SIGNAL SHUTDOWN`** command. Rather than rebooting, it might be better to have your system halt (shut down).

Change these two settings in the `/etc/inittab` file as follows:

- Modify `id:5:initdefault:` to `id:3:initdefault:`
- Modify **`shutdown -r`** to **`shutdown -h`**

```
# cd /etc
# cp inittab inittab.orig
# vi inittab
...
# The default runlevel is defined here
id:3:initdefault:
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

These changes will be picked up when the system is rebooted.

8.5.7 Verifying the On Demand Timer Patch is set

In previous versions of SLES, the kernel hz_timer defaulted to 1, which can negatively affect performance. For more discussion see the IBM Redbooks publication *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926, which is available on the Web at:

`http://www.redbooks.ibm.com/abstracts/sg246926.html`

Following is an excerpt:

> Traditionally, the Linux kernel keeps track of time using a timer that interrupts the kernel at a constant rate. On each interrupt, the global variable jiffies is incremented and various queues are inspected for work. On dedicated hardware, this has a minimal performance effect. However, when running Linux running on z/VM, this is not a good idea.
>
> Although this method of keeping track of time may not be the most efficient, the biggest problem for Linux on z/VM is that the Linux guest uses a little bit of the CPU cycles every 10 ms. This causes the z/VM scheduler to keep the guest in queue. Therefore, unused memory pages in the Linux virtual machine cannot be trimmed. With many Linux guests holding on to their working sets, z/VM memory fills up quickly.
>
> This problem was addressed with the so-called on demand timer patch. The patch is included in Linux on System z distributions like Novell SUSE SLES8, SLES9, SLES10, and Red Hat RHEL3 Update 1, RHEL4, RHEL5.
>
> This patch does away with the 10 ms timer tick and sets the timer to pop only when the kernel needs to wake up. Even though this does not make a real zero load idle Linux

Archived

guest, the periods between two timer ticks are normally long enough for z/VM to recognize the guest as idle (and start taking measures to trim memory pages if necessary).

To verify the hz_timer is off, invoke the following command:

```
# cat /proc/sys/kernel/hz_timer
0
```

A value of 0 shows that the hz_timer is not ticking.

8.5.8 Configuring SSH keys

SSH sessions are typically authenticated through passwords typed in from the keyboard. With SSH *key-based authentication*, sessions can be authenticated through public and private keys so that no password is needed. To accomplish this, the following must be true:

- The SSH server system must have the client's public key.
- The SSH client must send its private key.
- The keys must match cryptographically.

SSH key-based authentication can be set up from the controller (client) to the virtual servers. If the golden image has a copy of controller's public key in the file `/root/.ssh/authorized_keys`, and the controller has a symbolic link to its private key in the file `/root/.ssh/id_dsa`, then key-based authentication will work to the cloned virtual servers.

To set up key-based authentication from the controller to the golden image (and consequently from controller to all cloned images), perform the steps that follow. You will need ssh sessions on both the golden image (`virtc529`) and the controller (`virtc530`), so prefixes are used in the following examples.

- On the golden image, create the directory `/root/.ssh` and set the permission bits to octal 700 with the **chmod** command:

```
virtc529: # mkdir /root/.ssh
virtc529: # chmod 700 /root/.ssh
```

- Start an SSH session as root on the controller
- Copy the controller's public key from the `/etc/ssh/` directory to the `/root/.ssh/` directory that you just created on the golden image (9.12.5.29):

```
virtc530: # cd /etc/ssh
virtc530: # scp ssh_host_dsa_key.pub <9.12.5.29>:/root/.ssh/authorized_keys
The authenticity of host '9.12.5.29 (9.12.5.29)' can't be established.
RSA key fingerprint is 03:09:c1:b9:a3:9b:a6:55:95:fa:2d:17:79:17:13:d3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.5.29' (RSA) to the list of known hosts.
Password:
ssh_host_dsa_key.pub                      100%  603     0.6KB/s   00:00
```

- On the controller, create a symbolic link from `/root/.ssh/id_dsa` to the controller's private key:

```
virtc530: # cd /root/.ssh
virtc530: # ln -s /etc/ssh/ssh_host_dsa_key id_dsa
```

- Verify the symbolic link was created through the **ls -l** command:

```
# ls -l
lrwxrwxrwx 1 root root  25 May 20 13:19 id_dsa -> /etc/ssh/ssh_host_dsa_key
-rw-r--r-- 1 root root 658 May 20 13:17 known_hosts
```

Verify that you can **ssh** into the golden image without needing a password.

```
virtc530: # ssh <9.12.5.29>
Last login: Tue May 20 09:56:15 2008 from ibm-44ad95aeb23.pok.ibm.com
virtc529:~ #
```

If you get an SSH session without having to supply a password, as with the above example, it shows that key-based authentication is working.

8.5.9 Modifying zipl.conf

When a SLES10 SP2 system starts, there are two ways it can determine which disks to bring online.

- The default method is that the device addresses are built into the initial RAMdisk.
- The disks to bring online can also be written to the zipl configuration file, `/etc/zipl.conf`, after which the **zipl** command must be run.

We recommend that you use the latter method so it will be easier to determine which minidisks are designed to be part of the system. It is much easier to read a file (`/etc/zipl.conf`) than to find where the disks are specified in the initial RAMdisk.

Also add the **vmpoff=LOGOFF** parameter and set the timeout to 3 seconds, as you did on the controller.

Make a backup copy of the file and add the string **dasd=100-101,300-301** to the parameter line in the `[ipl]` section:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf   // add the dasd= string
# Modified by YaST2. Last modification on Wed Apr  9 19:02:36 UTC 2008
[defaultboot]
defaultmenu = menu


:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 3
    1 = ipl
    2 = Failsafe
    3 = ipl
    4 = Failsafe

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
    image = /boot/image
    target = /boot/zipl
    ramdisk = /boot/initrd,0x1000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-101,300-301
vmpoff=LOGOFF TERM=dumb"
###Don't change this comment - YaST2 identifier: Original name: failsafe###
[Failsafe]
    image = /boot/image-2.6.16.60-0.3-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.16.60-0.3-default,0x1000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1   TERM=dumb 3"
```

Now, run the **zipl** command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'ipl' (default)
Adding #2: IPL section 'Failsafe'
Preparing boot device: dasda ().
Done.
```

8.5.10 Cleaning temporary files

Now is the best time to clean temporary files from golden image. If they stay there, they will be copied to each clone and occupy space there. There are a number of directories that can be cleaned up, and the history of the commands executed can be cleared.

- Create a script **cloneprep.sh** that can be run to prepare for cloning:

```
# cd /usr/local/sbin
# vi cloneprep.sh
#!/bin/bash
rm -fr /tmp/.ICE-unix
rm -fr /tmp/.X11-unix
rm -rf /tmp/*
rm -f /var/log/YaST2/*
history -c
```

  You may choose different steps to prepare for cloning.

- Make the script executable through the **chmod +x** command:

```
# chmod +x cloneprep.sh
```

- Call the script interactively:

```
# cloneprep.sh
```

The system should now be cleaned for cloning.

8.5.11 Rebooting the system and verifying changes

You are now finished customizing the golden Linux image. Now **reboot** to test your changes:

```
# reboot
Broadcast message from root (pts/0) (Sun Nov 11 14:57:18 2007):

The system is going down for reboot NOW!
```

When the system comes back up, you should verify the changes that you made.

1. SSH back into the golden image and check a few settings.
2. Use the **df** command to display your file systems:

```
# df -h
/dev/dasda1           373M  110M  244M  32% /
udev                  247M  104K  247M   1% /dev
/dev/mapper/system--vg-opt--lv
                      372M   91M  262M  26% /opt
/dev/mapper/system--vg-tmp--lv
                      372M   17M  337M   5% /tmp
/dev/mapper/system--vg-usr--lv
                      2.0G  958M  957M  51% /usr
```

Archived

```
/dev/mapper/system--vg-var--lv
                      496M   67M  404M  15% /var
```

3. Confirm that both of your swap spaces are operational:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       259956  0       -1
/dev/dasdd1                             partition       519924  0       -2
```

   Devices with higher priority will be used first.

4. Shut down your golden image from the SSH session (to clone Linux, it should be shut down).

```
# shutdown -h now
```

At this point, you have successfully installed the golden image. This image will normally be shut down.

Archived

9

Cloning SLES 10 SP2

This chapter focuses on cloning the golden image. Examples are given for cloning manually and through the use of a script. The following steps are described:

- Formatting DASD for minidisks
- Defining a new user ID for a virtual server
- Cloning a virtual server manually
- Cloning a virtual server automatically
- Creating three more virtual servers

Archived

© Copyright IBM Corp. 2008. All rights reserved.

9.1 Formatting DASD for minidisks

In 4.6.2, “Formatting DASD for minidisks” on page 50, DASD was formatted to become minidisks for the controller and the golden image. The **CPFMTXA** command can be used to format one DASD at a time, but the **CPFORMAT EXEC** is a wrapper around **CPFMTXA** that allows the formatting of multiple DASDs.

To have access to enough DASD to define four more user IDs with two 3390-3 volumes each, seven more volumes must be formatted (eight are needed, but one volume, D957 in this example, is still available). In the examples used in this book, the seven volumes that will be formatted are at addresses DA51-DA57. Consult your worksheets on 2.7.2, “z/VM DASD worksheet” on page 17.

1. Logon to a 3270 session as MAINT.
2. Query the devices that will be used for LINUX00 (controller), LNXMAINT (common CMS files) and SLES10S2 (the golden image). In this example they are D856-D857 D952-D957.

```
==> q da51-da57
DASD DA51 NWDA51 , DASD DA52 NWDA52 , DASD DA53 NWDA53 , DASD DA54 NWDA54
DASD DA55 NWDA55 , DASD DA56 NWDA56 , DASD DA57 NWDA57
```

3. Attach the seven volumes that will be used for the controller, the common CMS disk and the golden image. In this example it is the DASD at the following addresses:

```
==> att da51-da57 *
DA51-DA57 ATTACHED TO MAINT
```

4. Invoke the **CPFORMAT** command against these volumes using the parameter **as perm**:

```
==> cpformat da51-da57 as perm

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA51 MAINT    DA51 3390  NWDA51 DA51          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA52 MAINT    DA52 3390  NWDA52 DA52          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA53 MAINT    DA53 3390  NWDA53 DA53          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA54 MAINT    DA54 3390  NWDA54 DA54          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA55 MAINT    DA55 3390  NWDA55 DA55          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA56 MAINT    DA56 3390  NWDA56 DA56          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA57 MAINT    DA57 3390  NWDA57 DA57          0       3339

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PERM space (y/n)?
y
...
DA51-DA57 DETACHED
DA51-DA57 ATTACHED TO MAINT

DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    DA51 MAINT    DA51 3390  MMDA51 DA51          0       3339
MAINT    DA52 MAINT    DA52 3390  MMDA52 DA52          0       3339
MAINT    DA53 MAINT    DA53 3390  MMDA53 DA53          0       3339
MAINT    DA54 MAINT    DA54 3390  MMDA54 DA54          0       3339
MAINT    DA55 MAINT    DA55 3390  MMDA55 DA55          0       3339
```

```
MAINT    DA56 MAINT    DA56 3390  MMDA56 DA56          0       3339
MAINT    DA57 MAINT    DA57 3390  MMDA57 DA57          0       3339
```

Detach the seven volumes from MAINT through the **DETACH** command:

```
==> det da51-da57
DA51-DA57 DETACHED
```

Attach the newly formatted DASD to SYSTEM so they can be used for minidisks:

```
==> att da51-da57 system
DASD DA51 ATTACHED TO SYSTEM MMDA51
DASD DA52 ATTACHED TO SYSTEM MMDA52
DASD DA53 ATTACHED TO SYSTEM MMDA53
DASD DA54 ATTACHED TO SYSTEM MMDA54
DASD DA55 ATTACHED TO SYSTEM MMDA55
DASD DA56 ATTACHED TO SYSTEM MMDA56
DASD DA57 ATTACHED TO SYSTEM MMDA57
```

The seven DASD volumes will now be available to be used for minidisks in the USER DIRECT file. They will also be available after the next IPL because their new labels match the pattern specified by the **User_Volume_Include MM*** statement in the SYSTEM CONFIG file.

9.2 Defining a new user ID for a virtual server

In this section you will define a new user ID, LINUX01.

1. Logon to MAINT and edit the USER DIRECT file to add a new user ID, LINUX01. Define it with the same sized 100 and 101 minidisks so that the golden image can be cloned to it.

   ```
   ==> x user direct c
   ```

2. Go to the bottom of the file and add the following six lines. In this example the user ID will be LINUX01 with a password of LNX4VM. Two 3390-3 DASD are used (about 4.5 GB total). In this example, the devices are at real addresses D957 and DA51:

   ```
   *
   USER LINUX01 LNX4VM 256M 1G G
    INCLUDE LNXDFLT
    OPTION APPLMON
    MDISK 100 3390 0001 3338 <MMD957> MR LNX4VM LNX4VM LNX4VM
    MDISK 101 3390 0001 3338 <MMDA51> MR LNX4VM LNX4VM LNX4VM
   ```

3. Add the new volume to the $ALLOC$ user ID so that cylinder 0 will not show up in the disk map as a gap. Save your changes with the **FILE** subcommand:

   ```
   ====> top
   ====> /alloc
   USER $ALLOC$  NOLOG
    ...
    MDISK A0B 3390 000 001 <MMD957> R
    MDISK A0C 3390 000 001 <MMDA51> R
   ====> file
   ```

4. Again, check for gaps and overlaps. You can use the **ALL** subcommand with the logical OR operator “|” to check for both strings. You should see only one 501 cylinder gap.

   ```
   ==> diskmap user
   ==> x user diskmap
   ====> all /gap/|/overlap/
   -------------------  4  line(s) not displayed --------------------
                                          0         500         501    GAP
   ```

```
--------------------  368  line(s) not displayed --------------------
====> quit
```

5. Bring the changes online with the **DIRECTXA** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 45 disk pages
```

The new Linux user ID has now been defined.

9.2.1 Adding LINUX01 to AUTOLOG1 PROFILE EXEC

The new Linux ID you defined needs access to the VSWITCH. A **SET VSWITCH** command with the **GRANT** parameter can be added to the AUTOLOG1 **PROFILE EXEC** to do this. Also, an **XAUTOLOG** statement can be added if the user ID is automatically logged on at z/VM IPL time.

Link and access the AUTOLOG1 191 disk read-write and edit the file **PROFILE EXEC**. Add LINUX01 to the sections that grant access to the VSWITCH and that automatically start the Linux user IDs:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
==> x profile exec f   // add two lines
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'                 /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'        /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant linux00'
'cp set vswitch vsw1 grant sles10s2'
'cp set vswitch vsw1 grant linux01'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog linux00'
'cp xautolog linux01'
'cp logoff'                         /* logoff when done */
====> file
```

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

```
==> set vswitch vsw1 grant linux01
Command complete
```

The user ID LINUX01 should now have permission to attach its NIC to the VSWITCH.

9.2.2 Creating a parameter file for the new LINUX ID

For each Linux guest you want to clone, you need to create a parameter file. This file specifies many of the installation parameters. It will be used both when cloning to this user ID, and when installing SLES10 manually.

1. Logon to LNXMAINT.
2. Copy an existing parameter file to a new one with the file name equal to the new Linux user ID.

```
==> copy sles10s2 parm-s10 d linux01 parm-s10 d
==> x linux01 parm-s10 d
```

3. Edit the new parameter file as you did for SLES10 (see 7.3, “Preparing SLES10 SP2 bootstrap files” on page 94). If the new Linux is going to be on the same network as the controller, you will likely only have to change two variables: the IP address and the DNS name. In this example, the IP address is set to 9.12.5.31 and the DNS name is set to virtc531.pbm.ihost.com:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=<9.12.5.31> Hostname=<virtc531.itso.ibm.com>
Gateway=9.12.4.1 Netmask=255.255.252.0
Broadcast=9.12.4.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7 Portname=dontcare
Install=nfs://9.12.4.185/install/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
UseVNC=1  VNCPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

4. Logoff from LNXMAINT and logon to LINUX01.
5. Watch for error messages as LINUX01 is logged on. Answer n to the question Do you want to IPL Linux from minidisk 100?.

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 08:58:08 EDT WEDNESDAY 05/21/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
...
```

6. Verify that the new Linux user ID has an NIC at addresses 600-602:

```
==> q osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
00:      0600 DEVTYPE OSA         CHPID 20 OSD
00:      0600 MAC 02-00-01-00-00-04 CURRENT
00:      0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
00: OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0003
00:      0601 DEVTYPE OSA         CHPID 20 OSD
00:      0601 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
00: OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 0004
00:      0602 DEVTYPE OSA         CHPID 20 OSD
```

Archived

```
00:      0602 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
Ready; T=0.01/0.01 08:59:37
```

7. Verify that the minidisks at 100 and 101 and the VDISKs at 300 and 301 are read-write:

```
==> q da
00: DASD 0100 3390 MMD957 R/W       3338 CYL ON DASD  D957 SUBCHANNEL = 0000
00: DASD 0101 3390 MMDA51 R/W       3338 CYL ON DASD  DA51 SUBCHANNEL = 0001
00: DASD 0190 3390 MVD850 R/O        107 CYL ON DASD  D850 SUBCHANNEL = 0009
00: DASD 0191 3390 MMD857 R/O        300 CYL ON DASD  D857 SUBCHANNEL = 000C
00: DASD 019D 3390 MVD853 R/O        146 CYL ON DASD  D853 SUBCHANNEL = 000A
00: DASD 019E 3390 MVD853 R/O        250 CYL ON DASD  D853 SUBCHANNEL = 000B
00: DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
00: DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
00: DASD 0592 3390 MVD854 R/O         70 CYL ON DASD  D854 SUBCHANNEL = 000D
```

8. Logoff from LINUX01. To clone to a user ID, it should be logged off so the target disks can be linked read-write.

You should now be ready to clone to this new user ID.

9.3 Cloning a virtual server manually

Before using the shell script **clone.sh** to clone a server, you may want to clone a server manually to better understand the process.

There are many ways to clone Linux under z/VM. The steps in this section are just one way to do it. The following assumptions are made based on what you have done so far:

- The source (golden) Linux image is on the user ID SLES10S2 on minidisks 100 and 101.
- The target user ID, LINUX01 in this example, has identically sized minidisks.
- The **vmcp** command is available to the controller, LINUX00, to issue z/VM CP commands.
- The z/VM **FLASHCOPY** command can be used but, if you do not have that support, the Linux **dasdfmt** and **dd** commands will work.

Given these assumptions, one set of steps that can be used to clone a system is as follows:

1. Linking the source disks read-only.
2. Linking the target disks read-write.
3. Copying the source to the target disk through **FLASHCOPY** or the Linux **dasdfmt** and **dd** commands.
4. Mounting the newly copied root file system.
5. Modifying the networking information about the target system.
6. Detaching the target disks.
7. IPLing the target system.
8. Modifying the SSH keys on the target system.

The following sections describe these steps in detail.

Archive

Linking the source disks read-write

Start an SSH session as root to the controller, LINUX00. The source minidisks at virtual addresses 100 and 101 are on the golden image, SLES10S2. They are linked read-only as virtual addresses 1100 and 1101 through the RR parameter to the CP **LINK** command:

```
# vmcp link sles10s2 100 1100 rr
# vmcp link sles10s2 101 1101 rr
```

Linking the target disk read-write

The target minidisks, also at addresses 100 and 101, are on the controller, LINUX01. They are linked multi-read (read-write if no other user ID has write access) through the MR parameter as virtual device 2100 and 2101 through the CP **LINK** command:

```
# vmcp link linux01 100 2100 mr
# vmcp link linux01 101 2101 mr
```

Copying the source to the target disk through FLASHCOPY

The two disks are copied through the CP **FLASHCOPY** command:

```
# vmcp flashcopy 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 END TO 2100 0 END
# vmcp flashcopy 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 END TO 2101 0 END
```

Archived

Note: If you do not have **FLASHCOPY** support, you can use the Linux **dasdfmt** and **dd** commands. You must first enable the source and target disks through the **chccwdev -e** command, and then determine the device name through the **lsdasd** command:

```
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
# chccwdev -e 1101
Setting device 0.0.1101 online
Done
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
# chccwdev -e 2101
Setting device 0.0.2101 online
Done
# lsdasd
0.0.0100(ECKD) at (94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(ECKD) at (94: 4) is dasdb : active at blocksize 4096, 543240 blocks, 2122 MB
0.0.0102(ECKD) at (94: 8) is dasdc : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0103(ECKD) at (94:12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at (94:16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0300(FBA ) at (94:20) is dasdf : active at blocksize 512, 524288 blocks, 256 MB
0.0.0301(FBA ) at (94:24) is dasdg : active at blocksize 512, 1048576 blocks, 512 MB
0.0.1100(ECKD) at (94:28) is dasdh : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.1101(ECKD) at (94:32) is dasdi : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.2100(ECKD) at (94:36) is dasdj : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.2101(ECKD) at (94:40) is dasdk : active at blocksize 4096, 600840 blocks, 2347 MB
```

In this example, the source devices are /dev/dasdh and /dev/dasdi and the target devices are /dev/dasdj and /dev/dasdk. Format the target disks through the **dasdfmt** command and then copy them through the **dd** command using a block size of 4 K (4096) bytes:

```
# dasdfmt -b 4096 -y -f /dev/dasdj
...
# dasdfmt -b 4096 -y -f /dev/dasdk
...
# dd if=/dev/dasdh of=/dev/dasdj bs=4096
600840+0 records in
600840+0 records out
# dd if=/dev/dasdi of=/dev/dasdk bs=4096
600840+0 records in
600840+0 records out
```

The golden image should now be copied to the target disks. Detach the target disks:

```
# chccwdev -d 2100
# vmcp det 2100
# chccwdev -d 2101
# vmcp det 2101
```

Mounting the newly copied root file system

Determine which device name is the target 100, or local 2100 disk:

```
# lsdasd | grep 2100
0.0.2100(ECKD) at (94:36) is dasdj : active at blocksize 4096, 600840 blocks, 2347 MB
```

The first target disk in this example is the device node /dev/dasdj. The first partition is the root file system; it will be /dev/dasdj1.

Mount it over the mount point /mnt and **cd** into it:

```
# mount /dev/dasdj1 /mnt
# cd /mnt
```

Observe that this appears to be a root file system:

```
# ls
.rnd  boot  etc   lib    lost+found  mnt  proc  sbin  success  tmp  var
bin   dev   home  lib64  media       opt  root  srv   sys      usr
```

Modifying networking information about the target system

In this example, the only two pieces of networking information that are modified are the IP address (from 9.12.5.29 to 9.12.5.31) and the host name (from virtc529 to virtc531).

The host name is changed in the file /mnt/etc/HOSTNAME:

```
# cd etc
# vi HOSTNAME
virtc531.itso.ibm.com
```

The host name is changed twice in the file /mnt/etc/hosts:

```
# vi hosts
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
...
127.0.0.2       virtc531.itso.ibm.com virtc531
```

The IP address is changed in the file /mnt/etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.0600:

```
# cd sysconfig/network/
# vi ifcfg-qeth-bus-ccw-0.0.0600
BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot"
IPADDR="9.12.5.31"
MTU="1500"
NETMASK="255.255.252.0"
NETWORK="9.12.4.0"
BROADCAST="9.12.7.255"
_nm_name='qeth-bus-ccw-0.0.0600'
```

Detaching the target disks

Change directory to /, and use the **sync** command to flush the disks and the **umount** command to unmount the modified root file system:

```
# cd /
# sync
# umount /mnt
```

Detach the target minidisks through the CP **DETACH** command:

```
# vmcp det 2100
DASD 2100 DETACHED
# vmcp det 2101
DASD 2101 DETACHED
```

The newly copied and modified system disks have now been detached.

IPLing the target system

You should now be ready IPL the manually cloned system.

Logon to LINUX01 and IPL from minidisk 100:

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 09:22:04 EDT WEDNESDAY 05/21/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
00: zIPL v1.6.3 interactive boot menu
00:
00:  0. default (ipl)
00:
00:  1. ipl
00:  2. Failsafe
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 3 seconds):
00: Booting default (ipl)...
Linux version 2.6.16.60-0.21-default (geeko@buildhost) (gcc version 4.1.2 200701
15 (SUSE Linux)) #1 SMP Tue May 6 12:41:02 UTC 2008
We are running under VM (64 bit mode)
...
```

Watch for error messages on the console. Your new system should come up cleanly using the modified IP address and host name.

Modifying the SSH keys on the target system

Start an SSH session as root to the new system, using the new IP address or host name.

The SSH keys that were copied are identical to those of the golden image. Manually create three new ones with the **ssh-keygen** command:

```
# cd /etc/ssh
# ssh-keygen -t rsa -N "" -q -f ssh_host_rsa_key
ssh_host_rsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t dsa -N "" -q -f ssh_host_dsa_key
ssh_host_dsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t rsa1 -N "" -q -f ssh_host_key
ssh_host_key already exists.
Overwrite (y/n)? y
```

At this point, you have now cloned a Linux system manually. You can look around the new system. It should be identical to the golden image except for the IP address and host name.

Archived

Next, you can learn how to clone a Linux system automatically. You will use the LINUX01 user ID again. To clone, the target user ID must be logged off. You could shut the new system down cleanly, but because you will be cloning again, it does not matter. Go to the 3270 session and crash the LINUX01 user ID through the **LOGOFF** command:

```
==> #cp log
```

9.4 Cloning a virtual server automatically

Now that you have cloned a server manually and better understand the steps, you can use the **clone.sh** script to clone automatically.

Start an SSH session as root to the controller. The **clone.sh** script should be in your PATH in the directory /usr/local/sbin/. You can verify this with the **which** command:

```
# which clone.sh
/usr/local/clone.sh
```

The script takes two pairs of parameters:

from <user ID> The source Linux

to <user ID> the target Linux system

Both the source and target user IDs must be logged off. The script reads the parameter file on the LNXMAINT 192 disk (the controller's 191 disk) to obtain information necessary to give the new Linux virtual server an identity. It calls CP **FLASHCOPY** through the **vmcp** module/command to try to copy the golden image's 100 and 103 disks.

If **FLASHCOPY** fails, the script falls back to copying the disk through the Linux **dasdfmt** and **dd** commands. The script then boots the new Linux through the **XAUTOLOG** command. It also creates an empty backup directory under /backup/linux/ and adds the server's public key to the controller's known_hosts file.

It should take less than a minute to clone with **FLASHCOPY** support, and 10 to 20 minutes without it. Following is an example of cloning to the LINUX01 user ID with **FLASHCOPY** support. The output is divided into sections.

```
# clone.sh from sles10s2 to linux01
Checking that SLES10S2 exists and is not logged on ...
Invoking CP command: QUERY SLES10S2
HCPCQU045E SLES10S2 not logged on
Error: non-zero CP response for command 'QUERY SLES10S2': #45
Checking that LINUX01 exists and is not logged on ...
Invoking CP command: QUERY LINUX01
HCPCQU045E LINUX01 not logged on
Error: non-zero CP response for command 'QUERY LINUX01': #45
Device already is offline
Done
Setting device 0.0.0191 online
Done
SLES10S2 PARM-S10 D1 V         72          9          1  5/20/2008  9:19:53
LINUX01  PARM-S10 D1 V         72          9          1  5/20/2008 16:23:14

WARNING!!: Mindisks 100 and 101 will be copied to LINUX01
New host name will be:  virtc531.itso.ibm.com
New IP address will be: 9.12.5.31
Other network data is retrieved from LINUX01 PARM-S10 on 191 disk
Are you sure you want to overwrite these disks (y/n): y
```

In the preceding section, it is verified that the user IDs SLES10S2 and LINUX01 exist and are logged off. The 191 disk is deactivated and reactivated to pick up any changes that may have occurred. It is then verified that the two parameter files, SLES10S2 PARM-S10 and LINUX01 PARM-S10 exist. Next, you are asked to confirm that the disks on the target system can be written to. Respond yes by entering y.

```
Linking source and target 100 disks ...
Invoking CP command: link SLES10S2 100 1100 rr

Invoking CP command: link LINUX01 100 2100 mr

Copying 100 disks ...

FLASHCOPYing 1100 to 2100 ...
Invoking CP command: FLASHCOPY 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 END TO 2100 0 END
Invoking CP command: det 1100
DASD 1100 DETACHED
Linking source and target 101 disks ...
Invoking CP command: link SLES10S2 101 1101 rr

Invoking CP command: link LINUX01 101 2101 mr

Copying 101 disks ...

FLASHCOPYing 1101 to 2101 ...
Invoking CP command: FLASHCOPY 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 END TO 2101 0 END
Invoking CP command: det 1101
DASD 1101 DETACHED
Invoking CP command: det 2101
DASD 2101 DETACHED
```

In the preceding section, the script copies the source 100 and 101 disks to the target user ID through the **FLASHCOPY** command. If **FLASHCOPY** is not supported or fails for some other reason, the script falls back to the **dasdfmt** and **dd** commands to copy the disks.

```
Mounting newly cloned image over /mnt/targetLinux ...
Setting device 0.0.2100 online
Done
Mounting /dev/dasdi1 over /mnt/targetLinux ...
Modifying cloned image under /mnt/targetLinux ...
Regenerating SSH keys in /mnt/targetLinux/etc/ssh/ ...
Adding 9.12.5.31 to known_hosts file
Setting device 0.0.2100 offline
Done
Invoking CP command: DETACH 2100
DASD 2100 DETACHED
```

In the preceding section, the script mounts the newly copied root file system over a mount point, /mnt/targetLinux/. The networking information is modified and the SSH keys are regenerated. Then the disk is detached.

```
Invoking CP command: XAUTOLOG LINUX01
Command accepted
Successfully cloned SLES10S2 to LINUX01
You should be able to ping 9.12.5.31 within one minute
```

In the final section, the target user ID is logged on through **XAUTOLOG**. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. The new system should be in the network in about 30 to 45 seconds.

Archived

Note: If the **clone.sh** script fails, you can add the **-v** flag to obtain more diagnostic information. Also verify the following items:

- The target user ID has been granted access to the VSWITCH.
- The parameter file is copied and set correctly on the LNXMAINT 192 disk.
- The target 100 and 101 minidisks are sized identically.

9.5 Creating three more virtual servers

So far you have installed Linux manually twice onto LINUX00 and SLES10S2. You have created a new user ID LINUX01 and cloned to it. Now it is time to clone three more times to have one system for each of the virtual servers described in the remaining chapters.

The following steps are involved:

- Defining three more user IDs
- Testing logging on to a new user ID
- Creating three new parameter files
- Granting user IDs access to VSWITCH

9.5.1 Defining three more user IDs

Define three more user IDs for Linux virtual servers in the USER DIRECT file named LINUX02 - LINUX04. You will need to use the DASD volumes you just formatted: two for each virtual server. You can repeat the definition of LINUX01 three times with the block copy **""3** prefix command. For example:

```
==> x user direct
====> /user linux01
...
""3   *
02142 USER LINUX01 LNX4VM 256M 1G G
02143  INCLUDE LNXDFLT
02144  OPTION APPLMON
02145  MDISK 100 3390 0001 3338 <MMD957> MR LNX4VM LNX4VM LNX4VM
""     MDISK 101 3390 0001 3338 <MMDA51> MR LNX4VM LNX4VM LNX4VM
```

This will create three more copies of the LINUX01 user definition. Modify them to have a user ID of LINUX02 - LINUX04, and give each correct DASD labels:

```
*
 USER LINUX02 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3338 <MMDA52> MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 0001 3338 <MMDA53> MR LNX4VM LNX4VM LNX4VM
 *
 USER LINUX03 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3338 <MMDA54> MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 0001 3338 <MMDA55> MR LNX4VM LNX4VM LNX4VM
 *
 USER LINUX04 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
```

Archived

```
MDISK 100 3390 0001 3338 <MMDA56> MR LNX4VM LNX4VM LNX4VM
MDISK 101 3390 0001 3338 <MMDA57> MR LNX4VM LNX4VM LNX4VM
```

Go to the top of the file and find the definition for the user $ALLOC$. Add dummy definitions for cylinder 0 of each of the new volumes and save the changes:

```
====> top
====> /alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 520RES R
...
 MDISK A0B 3390 000 001 <MMD957> R
 MDISK A0C 3390 000 001 <MMDA51> R
 MDISK A0D 3390 000 001 <MMDA52> R
 MDISK A0E 3390 000 001 <MMDA53> R
 MDISK A0F 3390 000 001 <MMDA54> R
 MDISK A10 3390 000 001 <MMDA55> R
 MDISK A11 3390 000 001 <MMDA56> R
 MDISK A12 3390 000 001 <MMDA57> R
...
====> file
```

Check for overlaps and the single gap. Use the **QUIT** command to exit the USER DISKMAP file:

```
==> diskmap user
==> x user diskmap
====> pre off
====> all /gap/|/overlap/
-------------------  4  line(s) not displayed --------------------
                                              0         500         501    GAP
-------------------  368  line(s) not displayed --------------------
====> quit
```

Bring the changes online with the **DIRECTXA USER** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
```

You have now created three new user IDs that can be cloned to.

9.5.2 Creating three new parameter files

A new parameter must be created for each user ID with the proper networking information. Link and access the LNXMAINT 192 disk read-write and create three new parameter files:

```
==> link lnxmaint 192 1192 mr
==> acc 1192 f
==> copy linux01 parm-s10 f linux02 = =
==> copy linux01 parm-s10 f linux03 = =
==> copy linux01 parm-s10 f linux04 = =
```

Edit each of the three files, replacing the appropriate network values. For example, in LINUX02 PARM-S10, only the TCP/IP address and DNS name need to be modified because all other network and other values are the same:

```
==> x linux02 parm-s10 f
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=<9.12.5.32> Hostname=<virtc532.itso.ibm.com>
Gateway=9.12.4.1 Netmask=255.255.252.0
Broadcast=9.12.4.255 Layer2=0
```

```
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7 Portname=dontcare
Install=nfs://9.12.4.185/install/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
UseVNC=1  VNCPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
====> file
==> x linux03 parm-s10 f
...
```

When you are finished, release and detach the `LNXMAINT 192` disk.

```
==> rel f (det
DASD 1192 DETACHED
```

9.5.3 Granting user IDs access to VSWITCH

Modify the `PROFILE EXEC` on `AUTOLOG1 191` to grant access to the VSWITCH for the three new user IDs and add **XAUTOLOG** commands so they will booted when the z/VM system IPLs.

Link and access the `AUTOLOG1 191` disk so the file can be modified from `MAINT`:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
```

Edit the **PROFILE EXEC**:

```
==> x profile exec f
...
/* Grant access to VSWITCH for each Linux user */
'CP SET VSWITCH VSW1 GRANT SLES10'
'CP SET VSWITCH VSW1 GRANT LINUX01'
'CP SET VSWITCH VSW1 GRANT LINUX02'
'CP SET VSWITCH VSW1 GRANT LINUX03'
'CP SET VSWITCH VSW1 GRANT LINUX04'

/* XAUTOLOG each Linux user that should be started */
'CP XAUTOLOG SLES10'
'CP XAUTOLOG LINUX01'
'CP XAUTOLOG LINUX02'
'CP XAUTOLOG LINUX03'
'CP XAUTOLOG LINUX04'
====> file
```

It is easiest to grant access to the new user IDs for the current z/VM session with the **SET VSWITCH** command:

```
==> set vswitch vsw1 grant linux02
Command complete
==> set vswitch vsw1 grant linux03
Command complete
==> set vswitch vsw1 grant linux04
Command complete
```

Verify that the user IDs have access with the **QUERY VSWITCH ACCESSLIST** command:

```
==> query vswitch vsw1 acc
...
Authorized userids:
      LINUX00  LINUX01  LINUX02  LINUX03  LINUX04  SLES10S2
      SYSTEM
  RDEV: 3024.P00 VDEV: 3024 Controller: DTCVSW2
  RDEV: 3028.P00 VDEV: 3028 Controller: DTCVSW1  BACKUP
```

Archived

You have now defined three new user IDs suitable for cloning the golden image to.

9.5.4 Testing logging on to a new user ID

You should now be able to logon to a new user ID and verify the integrity of the definitions. Logon to LINUX02 and you should first notice that a NIC is created, as well as two VDISKs:

```
LOGON LINUX02
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 10:11:05 EDT WEDNESDAY 05/21/08
z/VM V5.4.0    2008-05-13 14:27

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

If you forgot to grant access to the VSWITCH, you will see an error message at this point. Verify that you have OSA devices at addresses 600-602, and read-write DASD devices at addresses 100-102:

```
==> q osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
00:      0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
...
==> q da
00: DASD 0100 3390 MMDA52 R/W       3338 CYL ON DASD  DA52 SUBCHANNEL = 0000
00: DASD 0101 3390 MMDA53 R/W       3338 CYL ON DASD  DA53 SUBCHANNEL = 0001
00: DASD 0190 3390 MVD850 R/O        107 CYL ON DASD  D850 SUBCHANNEL = 0009
00: DASD 0191 3390 MMD857 R/O        300 CYL ON DASD  D857 SUBCHANNEL = 000C
00: DASD 019D 3390 MVD853 R/O        146 CYL ON DASD  D853 SUBCHANNEL = 000A
00: DASD 019E 3390 MVD853 R/O        250 CYL ON DASD  D853 SUBCHANNEL = 000B
00: DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
00: DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
00: DASD 0592 3390 MVD854 R/O         70 CYL ON DASD  D854 SUBCHANNEL = 000D
```

Logoff from LINUX02.

You have now cloned one Linux virtual server and defined three more user IDs that should be ready for cloning to. You will clone to some of these user IDs in the chapters that follow.

9.6 Reviewing system status

Now you can view your system from a DASD perspective, as shown in Figure 9-1 on page 159. If you have followed the instructions in all sections of this book, you should have used 25 3390-3 volumes: 10 for your z/VM system, 7 for the Linux controller and golden image and 8 for the Linux virtual servers. The possible role of administrators and users is suggested in this figure.

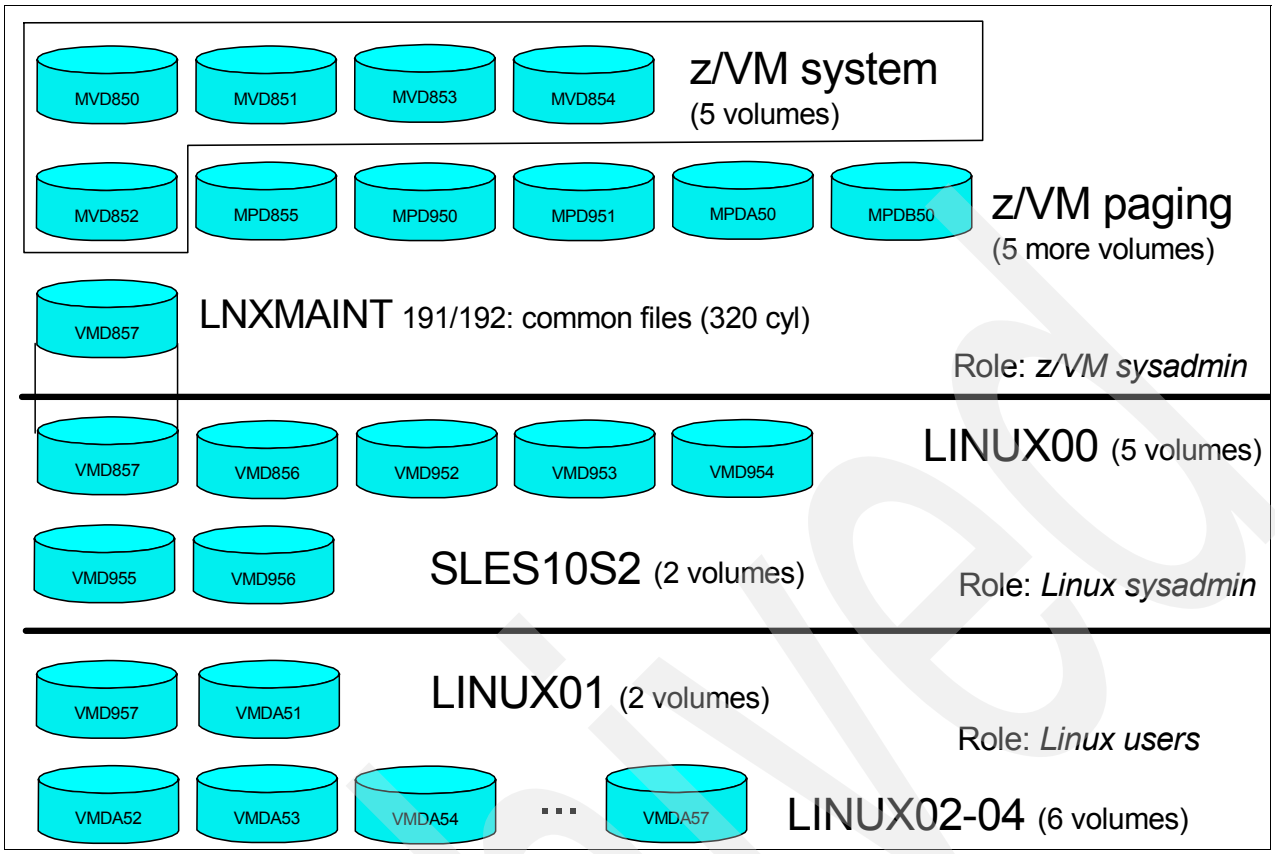


Figure 9-1 Linux virtual server system - DASD view and role view

These disk allocations can also be viewed by issuing the **QUERY DASD** command (from MAINT):

```
==> q da
DASD D850 CP OWNED  MVD850   62
DASD D851 CP OWNED  MVD851   1
DASD D852 CP OWNED  MVD852   0
DASD D853 CP OWNED  MVD853   65
DASD D854 CP OWNED  MVD854   27
DASD D855 CP OWNED  MPD855   0
DASD D856 CP SYSTEM MMD856   1
DASD D857 CP SYSTEM MMD857   3
DASD D950 CP OWNED  MPD950   0
DASD D951 CP OWNED  MPD951   0
DASD D952 CP SYSTEM MMD952   1
DASD D953 CP SYSTEM MMD953   1
DASD D954 CP SYSTEM MMD954   1
DASD D955 CP SYSTEM MMD955   0
DASD D956 CP SYSTEM MMD956   0
DASD D957 CP SYSTEM MMD957   1
DASD DA50 CP OWNED  MPDA50   0
DASD DA51 CP SYSTEM MMDA51   1
DASD DA52 CP SYSTEM MMDA52   0
DASD DA53 CP SYSTEM MMDA53   0
DASD DA54 CP SYSTEM MMDA54   0
DASD DA55 CP SYSTEM MMDA55   0
DASD DA56 CP SYSTEM MMDA56   0
DASD DA57 CP SYSTEM MMDA57   0
DASD DB50 CP OWNED  MPDB50   0
```

Archive

Archived

10

Cloning open source virtual servers

This chapter describes how to clone and customize the following Linux virtual servers:

- Creating a virtual Web server
- Creating a virtual LDAP server
- Creating a virtual file and print server
- Creating a virtual application development server

These Linux virtual servers can be thought of as *virtual appliances* after they have been cloned and *personalities* have been added to them.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

10.1 Creating a virtual Web server

The example in this section uses the LINUX01 user ID to create a virtual Web server. You should have a vanilla virtual server cloned to the user ID LINUX01 as described in Chapter 9, “Cloning SLES 10 SP2” on page 143.

10.1.1 Installing Apache RPMs

This section describes how to install the Apache Web server RPMs.

1. **SSH** into the IP address of the new LINUX01 server.
2. Install the following Apache RPMs through the **yast -i** command:

```
# yast -i apache2-prefork apache2 apache2-doc apache2-example-pages
```

3. You will see **yast** menus scroll by as the Apache RPMs are installed. When complete, you can confirm that the RPMs have been added through the **rpm -qa** command:

```
# rpm -qa | grep apache
apache2-doc-2.2.3-16.18
apache2-prefork-2.2.3-16.18
apache2-example-pages-2.2.3-16.18
apache2-2.2.3-16.18
```

10.1.2 Testing Apache

Start the Apache Web server to verify that it is installed successfully.

1. Start the Apache server through the **service** command and set it to start at boot time through the **chkconfig** command:

```
# service apache2 start
Starting httpd2 (prefork)                                  done
# chkconfig apache2 on
```

2. To verify that Apache is installed correctly, after it has been started, point a Web browser to the server and view the Apache test page. In your Web browser, put on the host name or IP address of your Web server as the URL. For example, the virtual server running on LINUX01 has a DNS name of virtc531.itso.ihost.com:

```
http://virtc531.itso.ibm.com
```

3. You should see a test page with the message: It Works!

If you receive an error when starting Apache, examine the log file /var/log/apache2/error-log for clues. If Apache started successfully but you cannot reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

10.1.3 Populating your Web site

You can begin to put your Web pages in the directory /srv/www/htdocs/, which is the default Web root. For security and customization purposes, you might want to change the default Web root to point to another directory. The easiest way to do this is to copy /etc/apache2/default-server.conf to your own configuration file, that is:

/etc/apache2/my-server.conf.

Archived

Make the changes in /etc/apache2/my-server.conf, and then edit /etc/apache2/httpd.conf to use my-server.conf.

10.1.4 Apache resources

The following Web sites contain additional information about Apache:

http://www.samspublishing.com/articles/article.asp?p=30115&seqNum=4
http://www.sitepoint.com/article/securing-apache-2-server-ssl
http://www.securityfocus.com/infocus/1786

10.2 Creating a virtual LDAP server

The Lightweight Directory Access Protocol (LDAP) is commonly implemented through the OpenLDAP package, which comes standard with most Linux distributions. Among other directory functions, OpenLDAP allows for centralized login authentication, and user ID and group ID resolution.

In this section you will clone Linux and configure a new virtual LDAP server. Then you will go back to the virtual Web server you just created and point it to the new LDAP server.

Next, you may want to configure the golden image so that it is pointing to this virtual server. If you do so, all Linux images that are cloned will be able to use this virtual LDAP server.

The steps in this section are as follows:

- Cloning a Linux
- Configuring the LDAP server
- Adding an LDAP user
- Setting another virtual server to use the LDAP server

10.2.1 Cloning a Linux

From a root session on the controller, clone from the golden image (SLES10S2 user ID) to LINUX02 through the **clone.sh** script:

```
# clone.sh from sles10s2 to linux02
...
```

You should now have a new virtual server running.

10.2.2 Configuring the LDAP server

To configure the OpenLDAP server, we recommend using the **yast** tool.

1. Start an SSH session as root to the new server.
2. Invoke the **yast** command. The YaST Control Center should appear:

```
# yast
  +--------------------------------------------------------------------------+
  ¦                           YaST Control Center                            ¦
  +--------------------------------------------------------------------------+

  +--------------------+ +-------------------------------------------------+
  ¦Software            ¦ ¦DNS and Hostname                                 ¦
  ¦Hardware            ¦ ¦Hostnames                                        ¦
```

```
¦System              ¦ ¦Kerberos Client                                ¦
¦Network Devices     ¦ ¦LDAP Browser                                   ¦
¦Network Services    ¦ ¦LDAP Client                                    ¦
¦Security and Users  ¦ ¦LDAP Server                                    ¦
¦Miscellaneous       ¦ ¦Mail Transfer Agent                            ¦
¦                    ¦ ¦NFS Client                                     ¦
...
```

3. Use the down arrow key to move to Network Services on the left side. Use the Tab or right arrow key to move to the right side and select **LDAP Server**. Press **Enter**. You should see the following message. Press **Enter** to accept the default of **Continue** and some RPMs will be installed:

```
These packages need to be installed: openldap2
```

4. The LDAP Server Configuration window should appear. Use the Tab key to move to **Yes** and press the Spacebar to select it, as shown in the following example. Move the cursor to the **Configure...** button through the Tab key and press **Enter**.

```
+Start LDAP Server-------------------------+
¦( ) No                                     ¦
¦                                           ¦
¦(x) Yes                                    ¦
¦                                           ¦
¦        [     Configure...       ]         ¦
¦                                           ¦
¦        [x] Register at an SLP Daemon      ¦
¦                                           ¦
+-------------------------------------------+
```

5. The LDAP Databases window should appear, displaying the text Available Databases:. There should be no databases listed. Move the cursor to the text **[Add Database...]** and press **Enter**.

6. You should see the Add Database window as shown in the following example. Set the Base DN (in this example, `dc=itso,dc=ibm,dc=com` is used), then set the LDAP administrator password (in this example, `lnx4vm`).

```
+-------------------------------------------+
¦                Add Database               ¦
¦                                           ¦
¦+General Settings-------------------------+¦
¦¦Base DN                                  ¦¦
¦¦dc=itso,dc=ibm,dc=com___________________¦¦
¦¦                                         ¦¦
¦¦Root DN                                  ¦¦
¦¦cn=Administrator________ [x] Append Base DN¦¦
¦¦                                         ¦¦
¦¦LDAP Password Validate Password Encryption ¦¦
¦¦******_______ ******___________ SSHA____v ¦¦
¦¦                                         ¦¦
¦¦Database Directory                       ¦¦
¦¦/var/lib/ldap/_________________ [Browse...]¦¦
¦+-----------------------------------------+¦
¦                                           ¦
¦+Password Policy Settings-----------------+¦
¦¦[ ] Enable Password Policies              ¦¦
¦¦[ ] Hash Clear Text Passwords             ¦¦
¦¦[ ] Disclose "Account Locked" Status      ¦¦
¦¦Default Policy Object DN                 ¦¦
¦¦<efault Password Policy_ [x] Append Base DN¦¦
¦+-----------------------------------------+¦
¦               [OK][Cancel]                ¦
+-------------------------------------------+
```

7. The LDAP Server Configuration window should appear again, this time with one database listed identified by the Base DN (dc=itso,dc=ibm,dc=com in this example). Move the cursor to **Finish** and press **Enter**.
8. The window Saving LDAP Server Configuration should appear. The database will be created and the LDAP server configured.
9. You should be returned to the YaST Control Center. Move the cursor to **Quit** and press **Enter**.
10. Verify that the LDAP server is running through the **service** command and that it is set to start in run levels 3 and 5 through the **chkconfig** command:

```
# service ldap status
Checking for service ldap:                                      running
# chkconfig --list ldap
ldap                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

You have now cloned a new virtual server and configured it to run OpenLDAP.

10.2.3 Adding an LDAP user

When the golden image was installed, it was recommended that a non-root user ID be added. In this example, it was named mikemac.

1. Verify that this user exists through the **id** command, and note that there is an entry in the /etc/passwd file through the **grep** command:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=100(users),16(dialout),33(video)
# grep mikemac /etc/passwd
mikemac:x:1000:100::/home/mikemac:/bin/bash
```

2. Delete this local user through the **userdel** command so it can be added to LDAP later.

```
# userdel mikemac
no crontab for mikemac
# id mikemac
id: mikemac: No such user

...
```

> **Important:** There appears to be a bug in YaST related to LDAP. An attempt was made to add users through **yast** -> **Security and Users** -> **User management** -> **LDAP Options** -> **LDAP User and Group Configuration**. Choosing this option opened a prompt window asking for the Administrator password. When the correct password was typed, an error message was issued stating A problem occurred while connecting to the LDAP server. Details of the message state Invalid Credentials.
>
> The following step works around this problem by adding a user through an LDIF file.

3. An LDAP Interchange Format (LDIF) file is created to create an organizational unit named People and a user ID named mikemac. Additional details are beyond the scope of this section.

```
# cat initial.ldif
dn: ou=People,dc=itso,dc=ibm,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit
```

```
dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac
```

4. Add the contents of the LDIF file to the LDAP server through the **ldapadd** command:

```
# ldapadd -x -h localhost -D "cn=Administrator,dc=itso,dc=ibm,dc=com" -f initial.ldif -W
Enter LDAP Password:
adding new entry "ou=People,dc=itso,dc=ibm,dc=com"

adding new entry "uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com"
```

5. Search for the new user ID just added through the **ldapsearch** command:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, itso.ibm.com
dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Archived

Note: The preceding output was created on a Beta-2 system. With the first GMC system (gold master candidate) system, the **ldapsearch** command did not appear to work:

```
# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 32 No such object

# numResponses: 1
```

The following **ldappasswd** command works correctly, so the data is clearly in the LDAP database.

This shows that the user ID exists in the LDAP database. Now you may want to set the password through the **ldappasswd** command:

```
# ldappasswd -x -D "cn=Administrator,dc=itso,dc=ibm,dc=com" -W -S
"uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com"
New password:
Re-enter new password:
Enter LDAP Password:
Result: Success (0)
```

At this point, you have added a new LDAP user through an LDIF file and set its password.

10.2.4 Setting another virtual server to use the LDAP server

Now that you have a virtual LDAP server, you may want to point another virtual server to it so that you will have a centralized user database. If you have been following along in this book, you have created a Web server running on the LINUX01 user ID. To point it to an LDAP server is fairly easy. In this section you will perform the following steps:

- Testing that the LDAP client is not working
- Using YaST to modify the LDAP authentication client
- Testing the LDAP client

Testing that the LDAP client is not working

Before you start, try a couple of commands to verify that LDAP is *not* working. Get an SSH session to the virtual Web server running on the user ID LINUX01. Use the LDAP user ID that you added earlier to the virtual LDAP server; it is mikemac, in this example.

```
# ldapsearch -x uid=<mikemac>
ldap_bind: Can't contact LDAP server (-1)
```

The **ldapsearch** command cannot resolve the LDAP user because it cannot contact the LDAP server.

The user `mikemac` is deleted from the local file system through the **`userdel`** command:

```
# userdel <mikemac>
no crontab for mikemac
```

Using YaST to modify the LDAP authentication client

The **`yast`** system administration interface can be used to configure the LDAP authentication client.

1. Invoke the **`yast`** command. The YaST Control Center should appear, as shown.
2. Select **Network Services** on the left side and **LDAP Client** on the right. Press **Enter**:

```
# yast
  +--------------------------------------------------------------------------+
  ¦                            YaST Control Center                           ¦
  +--------------------------------------------------------------------------+

  +--------------------+ +-------------------------------------------------+
  ¦Software            ¦ ¦DNS and Hostname                                ¦
  ¦Hardware            ¦ ¦Hostnames                                       ¦
  ¦System              ¦ ¦Kerberos Client                                 ¦
  ¦Network Devices     ¦ ¦LDAP Browser                                    ¦
  ¦Network Services    ¦ ¦LDAP Client                                     ¦
  ¦Security and Users  ¦ ¦LDAP Server                                     ¦
  ¦Miscellaneous       ¦ ¦Mail Transfer Agent                             ¦
  ¦                    ¦ ¦NFS Client                                      ¦
  ...
```

3. On the window that follows:
   a. Use the Tab key to move to **Use LDAP** and press the Spacebar to select that choice.
   b. Enter the IP address (or DNS name) of your LDAP server. You can either enter LDAP base DN manually or press **Fetch DN** and then **OK** in the result window. This way you can make sure that the LDAP server is accessible.
   c. Deselect LDAP TLS/SSL.

```
  +User Authentication---------------------------+
  ¦                                              ¦
  ¦ ( ) Do Not Use LDAP                          ¦
  ¦ (x) Use LDAP                                 ¦
  ¦ ( ) Use LDAP but Disable Logins              ¦
  ¦                                              ¦
  +----------------------------------------------+
  +LDAP Client-----------------------------------+
  ¦                                              ¦
  ¦ Addresses of LDAP Servers                    ¦
  ¦ 9.12.5.32_____________________________[Find] ¦
  ¦ LDAP Base DN                                 ¦
  ¦ dc=itso,dc=ibm,dc=com_____________[Fetch DN] ¦
  ¦ [ ] LDAP TLS/SSL                             ¦
  ¦ [ ] LDAP Version 2                           ¦
  +----------------------------------------------+
```

4. Use the Tab key to move to **Finish** and press **Enter**. You should get the following prompt. Press **Enter** to continue:

```
These packages need to be installed:
 pam_ldap
 nss_ldap
 pam_ldap-32bit
 nss_ldap-32bit
```

Archived

5. Accept **OK** if you get a warning window.
6. Press the **Quit** button on the main window to quit YaST.

Your Web server virtual Linux should now also be using OpenLDAP for user ID and group ID resolution and authentication.

Testing the LDAP client

Try the **id** command against the new LDAP user:

```
# id <mikemac>
uid=501(mikemac) gid=100(users) groups=100(users)
```

Note that the UID is 501 in this example (from the LDIF file), not the value 1000 (from the Linux installation). Try the **ldapsearch** command again:

```
# ldapsearch -x uid=mikem
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, itso.ibm.com
dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac
...
```

You should also be able to start an SSH session to the virtual Web server using the LDAP user.

You may also want to set the golden image to authenticate with the LDAP server. In this fashion, all virtual servers cloned after that will be able to utilize a centralized authentication server.

10.3 Creating a virtual file and print server

Samba allows Windows clients to map Linux file systems as shared drives. Samba can also act as a “middle man” between Windows clients and a Linux print server. The recommended Linux print server is the Common UNIX Printing System (CUPS). This section does not describe the configuration of CUPS, but it does describe how the necessary RPMs are installed.

The steps in this section are as follows:

- Cloning a Linux virtual server
- Installing the necessary RPMs

Archived

- Configuring the Samba configuration file
- Adding a Samba user
- Starting Samba at boot time
- Testing your changes

10.3.1 Cloning a Linux virtual server

Start an SSH session to the controller as root, and clone a basic virtual server. In this example, the user ID `LINUX03` is used.

```
# clone.sh from sles10s2 to linux03
...
```

SSH in to the new virtual server.

10.3.2 Installing the necessary RPMs

Add the following RPMs through the **`yast -i`** command:

```
# yast -i samba yast2-samba-server samba-doc samba-pdb samba-vscan samba-winbind cups
cups-drivers
```

You will see a number of YaST curses screens flash by as the RPMs are added to the system.

Confirm that the RPMs were added:

```
# rpm -qa | egrep "samba|cups"
cups-libs-1.1.23-40.41
samba-client-3.0.28-0.5
yast2-samba-server-2.13.24-0.3
samba-pdb-3.0.28-0.5
yast2-samba-client-2.13.40-0.3
samba-winbind-3.0.28-0.5
cups-1.1.23-40.41
samba-vscan-0.3.6b-42.71
cups-libs-32bit-1.1.23-40.41
cups-client-1.1.23-40.41
samba-doc-3.0.28-0.4
samba-3.0.28-0.5
cups-drivers-1.1.23-28.2
```

The Samba and CUPS RPMs are now installed.

10.3.3 Configuring the Samba configuration file

The one configuration file for Samba is `/etc/samba/smb.conf`. It is easy to add an SMB share that will be made available by the Samba server. A useful test directory is `/usr/share/doc/`, because it contains significant Linux documentation. The following example will create a file *share* named `sharedoc`:

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf    // add three lines at the bottom of the file:
...
[sharedoc]
        comment = SLES10 on System z documentation
        path = /usr/share/doc/
```

This will cause an SMB share named **sharedoc**, consisting of the contents of /usr/share/doc, to be created when Samba is started.

10.3.4 Adding a Samba user

The default method that Samba uses to determines users' credentials is to look in the /etc/samba/smbpasswd file. That user must first exist in the Linux file system (/etc/passwd, /etc/shadow, and so on).

The following example shows adding the user mikemac to the smbpasswd file. In this example, the user exists in the local file system. Perform the following steps:

1. Use the **id** command to verify the local user:

```
# id <mikemac>
uid=1000(mikemac) gid=100(users) groups=100(users),16(dialout),33(video)
```

2. To create a new Samba user, the **smbpasswd -a** command is used. If there are policy error messages, they can be ignored.

```
# smbpasswd -a <mikemac>
New SMB password:
Retype new SMB password:
account_policy_get: tdb_fetch_uint32 failed for field 1 (min password length),
...
Added user mikemac.
```

3. You can see that the last **smbpasswd** command added mikemac to the file smbpasswd:

```
# grep <mikemac> smbpasswd
...
mikemac:1000:F8BC610F17A8B726E72C57EF50F76A05:5FFEA84B2845CECE6FBEB2ECFF4F25D6:[U
]:LCT-48343DBD:
```

This method of maintaining Samba users, groups and passwords is useful for a small number of users. For a larger number of users, we recommend merging Samba and LDAP. It is not a simple as pointing the virtual file and print server at the virtual LDAP server as described in 10.2.4, "Setting another virtual server to use the LDAP server" on page 167, because the Samba schema must first be added to LDAP.

Details about that topic are beyond the scope of this book, but there are related presentations, *Directory Serving Solutions Using OpenLDAP*, and *File Serving Solutions Using Samba-3*, available on the Web at:

http://linuxvm.org/present/

10.3.5 Starting Samba at boot time

Samba consists of two daemons, **nmbd** and **smbd**. They can be started for the current session through the **service** command for both the **nmb** and **smb** daemons:

```
# service nmb start
Starting Samba NMB daemon                                  done
# service smb start
Starting Samba SMB daemon                                  done
```

The following **chkconfig** commands will set these daemons to start at boot time:

```
# chkconfig nmb on
# chkconfig smb on
```

Samba should now be running and configured to start at boot time.

10.3.6 Testing your changes

You can verify that the Samba daemons are running through the **service** command:

```
# service nmb status
Checking for Samba NMB daemon                                  running
# service smb status
Checking for Samba SMB daemon                                  running
```

You can test getting a Samba share from a Windows desktop.

1. Go to any Windows Explorer window (such as My Computer) and select **Tools** -> **Map Network Drive**.
2. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 10-1. In this example, the UNC is `\\9.12.5.33\sharedoc`.
3. You may need to click **Connect using different user name**, if the sample user ID and password are different on your desktop computer from the values you set on the Samba server.
4. Click **Finish**. If all the steps were correct, you should see the files in a new Explorer window as shown in the bottom right corner of Figure 10-1.

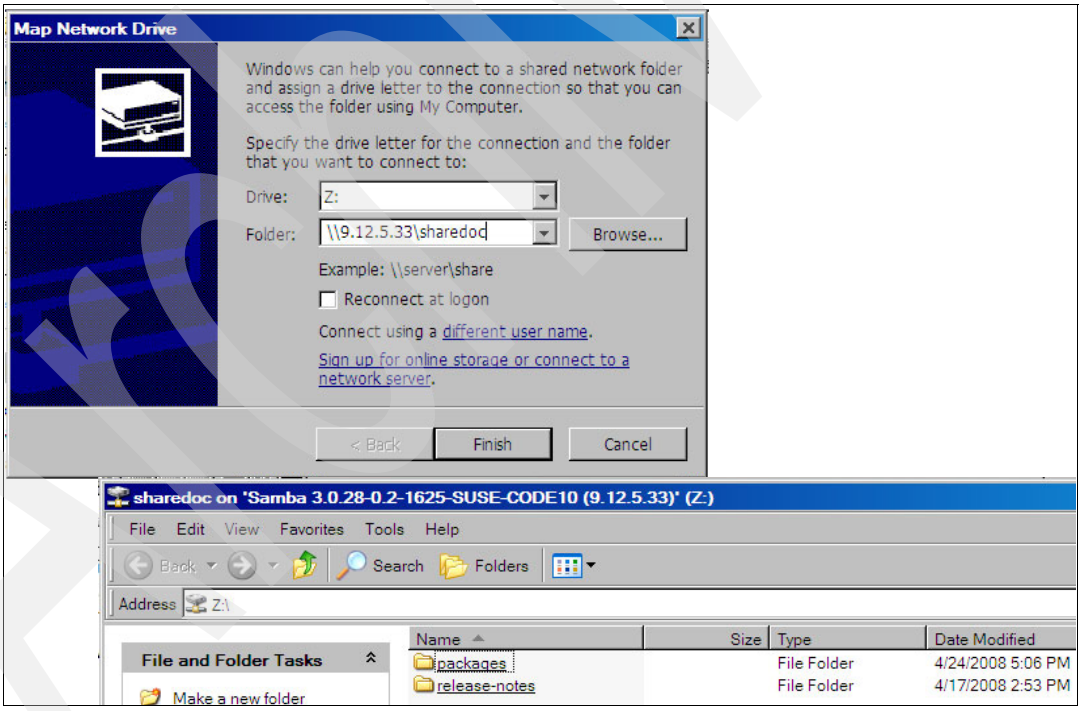


Figure 10-1 Mapping a network drive to the Samba server

You should now have Samba configured and running, with one new share available.

If you prefer to use the command line, you can map a network drive with the DOS **NET USE** command:

```
c:\> net use * \\<9.12.5.33>\sharedoc
Drive Y: is now connected to \\9.12.5.33\sharedoc.
```

Note: You may be prompted for the Samba user ID and password.

Another useful DOS command to interrogate the Samba server is the **NET VIEW** command. For example:

```
The command completed successfully.
c:\> net view \\9.12.5.33
Shared resources at \\9.12.5.33

Samba 3.0.28-0.5-1657-SUSE-CODE10

Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
groups      Disk           All groups
profiles    Disk           Network Profiles Service
sharedoc    Disk           SLES10 on System z documentation
users       Disk           All users
The command completed successfully.
```

10.3.7 Configuring printing

Configuring printing is more complex and is beyond the scope of this section. For more information about this topic, refer to the IBM Redpaper publication *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864:

http://www.redbooks.ibm.com/abstracts/redp3864.html?Open

10.4 Creating a virtual application development server

Most Linux distributions come with a basic set of application development tools, making Linux one of the most versatile development systems. These basic tools are ideal for projects of any size.

There are three main areas of development in Linux:

- Linux kernel development (C) for the Linux operating system itself, such as subsystems, device drivers, memory management
- Application development (C/C++ and Java) for software to be used on Linux
- Web development for applications to be run on the Web, such as stock trade applications or e-mail applications

The development languages used in implementation range from scripting languages such as Python or Tcl, to compiled languages such as C/C++ and Java. There is software available on Linux, for example MySQL™ and Apache, to help form a development system for developers to create integrated applications.

A popular open source Web platform is LAMP, which stands for the open source software and programming languages used to make up the platform: Linux, Apache, MySQL, Python or PHP. Other times, it is just as useful to know about Linux development tools when you want to build an application from source code downloaded from `www.sourceforge.net`.

Archived

10.4.1 Cloning a Linux virtual server

From the controller, clone a basic virtual server to LINUX04.

```
# clone.sh from sles10s2 to linux04
...
```

SSH in to the new virtual server as root.

You can install *all* the application development tools described in this section with the following command:

```
# yast -i python perl tcl php gcc gdb make java-1_5_0-ibm java-1_5_0-ibm-devel
java-1_5_0-ibm-fonts
```

These RPMs use approximately 100 MB in the /usr/ file system.

If you want to install only certain tools, each specific RPM or group of RPMs is described in the sections that follow.

10.4.2 Scripting languages

Scripts are good for quickly automating a process or writing your own commands. They are also used as the backbone of robust applications. There are numerous scripting languages used in Linux application development, and in this section we present overviews of the most popular and general ones, as obtained from their package descriptions.

- Python

  An interpreted, object-oriented programming language. For more information, refer to:

  http://www.python.org/

  To install the python interpreter, execute the command:

  ```
  # yast -i python
  ```

- perl

  Practical Extraction and Report Language (Perl) is optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is also useful for many system management tasks. For more information, refer to:

  http://www.perl.com/

  To install perl, execute the command:

  ```
  # yast -i perl
  ```

- tcl

  Tool Command Language (tcl), a simple programming language, provides basic language features such as variables, procedures, and control. It runs on most modern operating system, including UNIX, Macintosh, and Windows 95/98/NT computers. For more information, refer to:

  http://www.tcl.tk/

  To install tcl, execute the command

  ```
  # yast -i tcl
  ```

Archived

- PHP

  PHP: Hypertext Preprocessor (PHP), a widely-used open source general purpose scripting language that is especially suited for Web development, can be embedded into HTML. Although focused on server-side scripting, you can do much more with it. For more information, refer to:

  http://www.php.net/

  To install PHP, execute the command:

```
# yast -i php
```

10.4.3 C/C++ development tools

Most Linux distributions come with the C/C++ compiler, `gcc`. This is also known as the “GNU compiler collection” because it can compile other languages such as Fortran, but it is most frequently used to compile C and C++ code. In the minimal SLES10 installation, none of the development packages are installed. In order to use `gcc`, you must install it using yast:

```
# yast -i gcc
# rpm -qa | grep gcc
libgcc-4.1.2_20070115-0.21
gcc-4.1.2_20070115-0.21
```

gcc performs preprocessing, compilation, assembly and linking for files with extensions `.c` and `.cpp`, as well as numerous others (see the gcc manual page). Most C/C++ programs will require preprocessing, compilation, and assembly first to create object files; then linking combines all the object files into an executable file.

For security reasons, you should not use root for application development. You should either get another session as a non-root user, or from root, **su** to a non-root user. In this example, the non-root user `developer1` is used. The files `readfile.c` and `writefile.c` are compiled into the executable files:

```
# su - <developer1>
$ gcc -O -Wall -I/usr/local/include -o readfile.o -c readfile.c
$ gcc -O -Wall -I/usr/local/include -o writefile.o -c writefile.c
$ gcc -o fileoperations readfile.o writefile.o
```

The **-O** option is to generate optimized code, **-Wall** is used to display all warnings. The option **-I** is used to include header files needed by the source, and **-c** is to tell **gcc** not to run the linker. The last command links the two object files into one executable file. For debugging using **gdb**, you can generate symbolic information using the **-g** option:

```
$ gcc -g -O -Wall -I/usr/local/include -o readfile.o readfile.c
$ gcc -g -O -Wall -I/usr/local/include -o writefile.o writefile.c
$ gcc -g -o fileoperations readfile.o writefile.o
```

The GNU debugger, or **gdb**, is a very popular and robust debugger for C/C++ programs. You can step through your program (which has been successfully compiled) to see where it is failing. Install it using yast:

```
# yast -i gdb
```

A valuable tutorial is available that explains how to get started with gdb:

http://www.unknownroad.com/rtfm/gdbtut/gdbuse.html

Keep in mind that you can also set breakpoints at functions in the code. Refer to the manual page of gdb for more information: **man gdb**.

To make a large program more manageable, developers usually create a makefile that specifies instructions about how to compile a program. Then use the GNU **make** tool to use the makefile to make a working program. For more information about makefiles, see

http://vertigo.hsrl.rutgers.edu/ug/make_help.html

To install `make`, issue the command:

```
# yast -i make
```

10.4.4 Java development tools

SLES 10 SP2 comes with IBM Java Standard Development Kit (SDK) which is needed if you want to develop Java applications. You need a Java Runtime Environment (JRE™) if you only want to run Java applications. The following command installs both:

```
# yast -i java-1_5_0-ibm java-1_5_0-ibm-devel java-1_5_0-ibm-fonts
```

Again, you should perform application development as a non-root user. Open another SSH session and log in as a non-root user, or from the current session, **su** to a non-root user. Java programs are compiled using `javac`, as shown in this example:

```
# su - <developer1>
$ javac HelloWorld.java
$ java HelloWorld
Hello World!
```

The resulting file is `HelloWorld.class`, which can be run if there is a `main` method defined in `HelloWorld.java`. For Java applets, run with **appletviewer**. If you are using methods from other classes that are not in the same package, you can reference them using the **-classpath** option. For debugging information, use the **-g** option.

If your executable program has multiple class files, you can save the user time by making an executable `.jar` file. All you need to do is specify which class has the executable main method. This way the user simply needs one `.jar` file, instead of numerous class files.

First, create a manifest file that specifies where the main method is by adding a simple one-line command. You must specify the package name as well:

```
$ vi mainClass
Main-Class: myHello.HelloWorld
```

Now use the **jar** command to create the executable jar file that knows where the main method is. All of the needed files are in the directory `myHello`, and are in the package `myHello`. The **jar** command (similar to **tar**) packages the directory contents into one file with the **cmf** flags. After packaging, check the contents with the **tf** flags:

```
$ jar cmf mainClass myHello.jar myHello
$ jar tf myHello.jar
META-INF/
META-INF/MANIFEST.MF
myHello/
myHello/HelloWorld.java
myHello/HelloWorld.class
myHello/PrintScreen.java
myHello/PrintScreen.class
```

Run the `.jar` file by invoking **java** with the **-jar** option.

```
$ java -jar myHello.jar
Hello World!
```

A valuable Java debugger is **jdb**, it comes with IBMJava2-SDK and can be run similar to **gdb**. A useful tutorial is available on the Web at:

http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html

You can use the GNU **make** to build from Java makefiles, or use the more recent and popular Ant. Ant uses XML technology. A useful guide is available to get you started with either tool:

http://www.onlamp.com/pub/a/onlamp/2004/11/18/gnumake_3e.html

Your application development server is now ready to use.

10.4.5 Additional resources

The following Web sites are resources for additional information about application development topics:

Scripting languages

http://www.perl.com/
http://www.python.org/
http://www.freeos.com/guides/lsst/

C/C++

http://gcc.gnu.org/onlinedocs/gcc/
http://en.wikipedia.org/wiki/GNU_Compiler_Collection#External_links
http://vertigo.hsrl.rutgers.edu/ug/make_help.html

Java

http://www-130.ibm.com/developerworks/java/
http://java.sun.com/
http://csdl.ics.hawaii.edu/~johnson/613f99/modules/04/jar-files.html
http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html

Linux kernel development

http://www.kernel.org/pub/linux/docs/lkml/#blkd

Web development

http://www.onlamp.com/
http://cgi.resourceindex.com/
http://www.perl.com/

Help with vi

http://www.freeos.com/guides/lsst/misc.htm#commonvi

At this point, you have cloned and installed four virtual Linux *appliances*.

Archived

Archived

11

Miscellaneous recipes

This chapter contains the following sections describing miscellaneous tasks that you might want to perform:

- Adding a logical volume
- Extending an existing logical volume
- Centralizing home directories for LDAP users
- Setting the software clock through NTP
- Rescuing a Linux system
- Configuring rsync

Archived

© Copyright IBM Corp. 2008. All rights reserved.

11.1 Adding a logical volume

There are times when you require more disk space than a single direct access storage device (DASD) volume provides. For example, if you want to have a shared /home/ directory. you will want it to be of sufficient size. When this is the case, you can use the Logical Volume Manager (LVM) to combine multiple DASD volumes into one logical volume.

The following process describes how to create a logical volume with additional DASD on a Linux guest. The overall steps in adding a logical volume are:

- Adding minidisks to the z/VM directory entry
- Making the new minidisks available
- Formatting and partitioning the minidisks
- Creating a logical volume and file system
- Updating the file system table

11.1.1 Adding minidisks to the z/VM directory entry

The overall steps are described here. Specific details of these steps on z/VM are not given, but they are given for Linux. Use the LINUX02 user ID for this test.

Following are the high level steps to add two new 3390-3-sized minidisks to LINUX02:

- Determine the labels of the two volumes that will be added.
- Add minidisk statements to define minidisks (at virtual addresses 102 and 103 of size 3338 cylinders each, in this example) to the appropriate Linux user ID definition in the USER DIRECT file.
- Create the USER DISKMAP file to verify the disk layout.
- Bring the changes online with the **DIRECTXA** command.
- Shutdown the Linux system.
- Logoff the user ID.
- Log back on to it and IPL Linux.

11.1.2 Making the new minidisks available

When your system comes back up, start an SSH session to it. Use the **lsdasd** command to verify that the new minidisks are not seen yet (because there is dasd=100-102 in /etc/zipl.conf):

```
# lsdasd
0.0.(ECKD) at ( 94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(ECKD) at ( 94: 4) is dasdb : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0300(FBA ) at ( 94: 8) is dasdc : active at blocksize 512, 524288 blocks, 256 MB
0.0.0301(FBA ) at ( 94:12) is dasdd : active at blocksize 512, 1048576 blocks, 512 MB
```

Bring disks 102 and 103 online through the **chccwdev** command and verify they are available through the **lsdasd** command:

```
# chccwdev -e 102
Setting device 0.0.0102 online
Done
# chccwdev -e 103
Setting device 0.0.0103 online
Done
```

```
# lsdasd
0.0.(ECKD) at ( 94:  0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(ECKD) at ( 94:  4) is dasdb : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0300(FBA ) at ( 94:  8) is dasdc : active at blocksize 512, 524288 blocks, 256 MB
0.0.0301(FBA ) at ( 94: 12) is dasdd : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0103(ECKD) at ( 94: 16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0102(ECKD) at ( 94: 20) is dasdf : active at blocksize 4096, 600840 blocks, 2347 MB
```

You should be ready to format the new minidisks.

11.1.3 Formatting and partitioning the minidisks

You could format the minidisks sequentially, but you can also use the following bash **for** loop to put two **dasdfmt** jobs in the background to format both minidisks in parallel:

```
# for i in e f
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 2713
[2] 2714
```

When the jobs are finished you should see messages similar to the following:

```
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok
```

Now, use the **fdasd** command with the **-a** flag to create a single partition from each minidisk:

```
# fdasd -a /dev/dasde
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
# fdasd -a /dev/dasdf
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The minidisks are now ready for you to use in the logical volume. If you are creating a new logical volume, see 11.1.4, “Creating a logical volume and file system” on page 181. If you are extending an existing logical volume, skip ahead to 11.2, “Extending an existing logical volume” on page 186.

11.1.4 Creating a logical volume and file system

The overall steps involved in creating a logical volume are:

- Create physical volumes from the two partitions
- Create a single volume group
- Create a single logical volume
- Make a file system from the logical volume

Figure 11-1 on page 182 shows a block diagram of the logical volume manager reflecting this example.

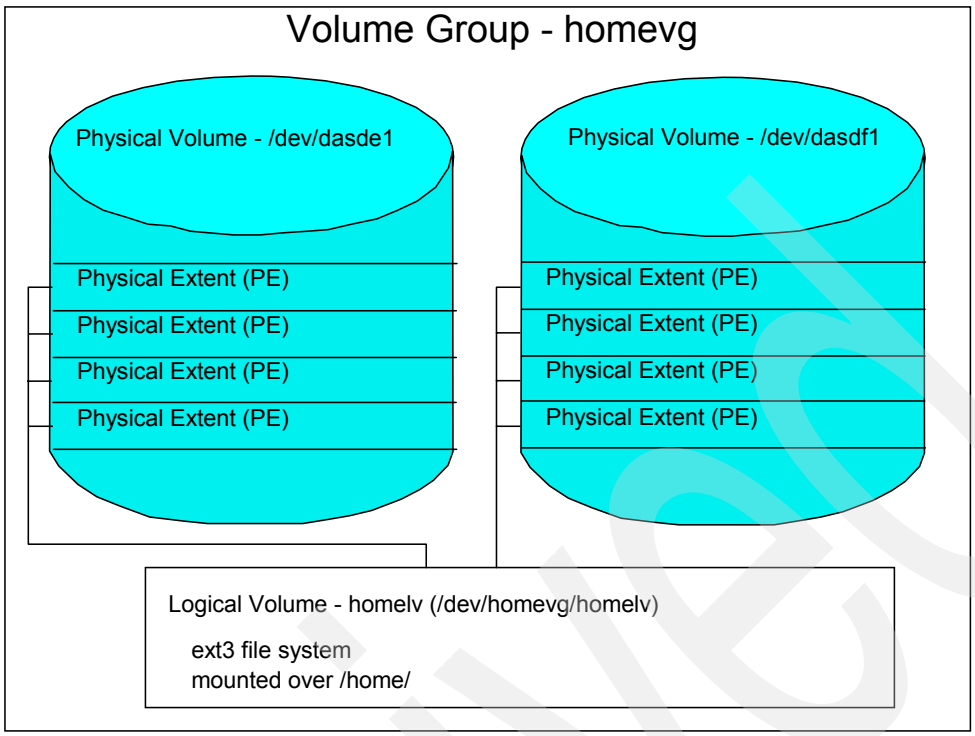


Figure 11-1 LVM block diagram

Creating physical volumes from the two DASDs

The **pvcreate** command initializes partitions for use by LVM. Initialize the two new DASD partitions.

```
# pvcreate /dev/dasde1 /dev/dasdf1
  Physical volume "/dev/dasde1" successfully created
  Physical volume "/dev/dasdf1" successfully created
```

Verify that the physical volumes were created through the **pvdisplay** command:

```
# pvdisplay
--- Physical volume ---
  PV Name               /dev/dasda2
  VG Name               system-vg
  ...
  --- Physical volume ---
  PV Name               /dev/dasdb1
  VG Name               system-vg
  ...
  --- NEW Physical volume ---
  PV Name               /dev/dasde1
  VG Name
  PV Size               2.29 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               8wJICO-Y5nP-TlOT-m721-QuTk-x6kY-MGqMAZ

  --- NEW Physical volume ---
  PV Name               /dev/dasdf1
```

```
  VG Name
  PV Size               2.29 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               DsKDtm-M8Ye-5shy-nfbV-n9mF-btGw-xef6WU
```

Creating a single volume group

The **vgcreate** command can be used to create a volume group named homevg from the two partitions. Use the **vgdisplay homevg** command to verify 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               4.58 GB
  PE Size               4.00 MB
  Total PE              1172
  Alloc PE / Size       0 / 0
  Free  PE / Size       1172 / 4.58 GB
  VG UUID               XHdfZg-wHNg-Iqpm-lvzz-XGMW-JEMk-UBdgCU
```

In this example, there are 1172 free physical extents.

Creating a single logical volume

The **lvcreate** command is used to create a logical volume. The **-l** flag specifies to use all free extents, 1172 in this example. The **-n homelv** specifies the name of the new logical volume. The last argument, **homevg**, specifies the name of the volume group from which the logical volume will be created.

```
# lvcreate -l 1172 -n homelv homevg
  Logical volume "homelv" created
```

Use the **lvdisplay** command to verify. The parameter is the full path of the logical volume, not just the logical volume name:

```
# lvdisplay /dev/homevg/homelv
  --- Logical volume ---
  LV Name                /dev/homevg/homelv
  VG Name                homevg
  LV UUID                f7Ipvk-nshF-KHeX-pRcf-cFYw-IpSU-aflyLv
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                4.58 GB
```

Archived

```
Current LE            1172
Segments              2
Allocation            inherit
Read ahead sectors    0
Block device          253:4
```

Making a file system from the logical volume

Now you have a logical volume. Use the **mke2fs** command to create a file system out of it. The **-j** flag adds a journal so it will be of type ext3:

```
# mke2fs -j /dev/homevg/homelv
mke2fs 1.38 (23-Jul-2007)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
600288 inodes, 1200128 blocks
60006 blocks (5.00%) reserved for the super user
...
```

The file system created from the logical volume is now ready to be mounted.

11.1.5 Updating the file system table

You could now 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. Make a backup copy, and then add the following line to the file:

```
# cd /etc
# cp fstab fstab.orig
# vi fstab
/dev/dasda1         /                  ext3       acl,user_xattr        1 1
/dev/system-vg/opt-lv /opt              ext3       acl,user_xattr        1 2
/dev/system-vg/tmp-lv /tmp              ext3       acl,user_xattr        1 2
/dev/system-vg/usr-lv /usr              ext3       acl,user_xattr        1 2
/dev/system-vg/var-lv /var              ext3       acl,user_xattr        1 2
/dev/disk/by-path/ccw-0.0.0300-part1 swap swap     defaults              0 0
/dev/disk/by-path/ccw-0.0.0301-part1 swap swap     defaults              0 0
/dev/homevg/homelv   /home              ext3       defaults              0 0
proc                /proc               proc       defaults              0 0
sysfs               /sys                sysfs      noauto                0 0
debugfs             /sys/kernel/debug   debugfs    noauto                0 0
devpts              /dev/pts            devpts     mode=0620,gid=5       0 0
```

Before mounting over `/home/`, check that it is empty. In this example a non-root user was created when installing the golden image, so that user has a directory under `/home/`. If you overmount `/home/`, this directory will be *covered*. Move it temporarily to another location; in this example, `/tmp/` is used.

```
# ls /home
mikemac
# mv /home/mikemac /tmp
# ls /home
```

Mount the `/home/` file system with one argument. Use the **`ls`** command to verify that there is no data in the logical volume except the `lost+found/` directory (created through the mke2fs

command). Move the existing home directory back from /tmp/. Use the **df -h** command to verify that it is mounted:

```
# mount /home
# mv /tmp/mikemac/ /home
# ls /home
lost+found  mikemac
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           373M  110M  244M  32% /
udev                  121M  124K  121M   1% /dev
/dev/mapper/system--vg-opt--lv
                      372M   91M  262M  26% /opt
/dev/mapper/system--vg-tmp--lv
                      372M   17M  337M   5% /tmp
/dev/mapper/system--vg-usr--lv
                      2.0G  958M  957M  51% /usr
/dev/mapper/system--vg-var--lv
                      496M   73M  399M  16% /var
/dev/mapper/homevg-homelv
                      4.6G  138M  4.2G   4% /home
```

When /home/ can be mounted with one parameter, it shows that the syntax in /etc/fstab is correct.

Now modify the **/etc/zipl.conf** file so the new disks are available the next time the system is rebooted. First make a backup of the working copy:

```
# cd /etc
# cp zipl.conf zipl.conf.works
# vi zipl.conf   // change the dasd= string
# Modified by YaST2. Last modification on Thu Apr 17 19:00:06 UTC 2008
[defaultboot]
defaultmenu = menu


:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 3
    1 = ipl
    2 = Failsafe
    3 = ipl
    4 = Failsafe

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
    image = /boot/image
    target = /boot/zipl
    ramdisk = /boot/initrd,0x1000000
   parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-101,300-301,102-103
vmpoff=LOGOFF TERM=dumb"
...
```

Note that the 100, 101, 300 and 301 disks are left in their existing locations. This is so they will be assigned the same device names at boot time.

Now, run the **zipl** command so the changes are written to the boot record:

```
# zipl
```


Archived

```
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'ipl' (default)
Adding #2: IPL section 'Failsafe'
Adding #3: IPL section 'ipl'
Adding #4: IPL section 'Failsafe'
Preparing boot device: dasda ().
Done.
```

Test a reboot to verify the new logical volume is successfully mounted over /home/.

```
# reboot

Broadcast message from root (pts/0) (Sat Apr 26 14:03:16 2008):

The system is going down for reboot NOW!
```

Verify that the logical volume comes back up and is mounted over /home/.

11.2 Extending an existing logical volume

This section describes the process of adding a new minidisk to an existing LVM. This is useful when your logical volume has run out of space.

First, repeat the steps at the beginning of this section to add a new minidisk. In this example, a minidisk at virtual address 104 is added of size 3338 cylinders. Remember to logoff and log back on to LINUX02 so the new directory entry is read.

When your system comes back, bring disk 104 online and verify that it is available:

```
# chccwdev -e 104
Setting device 0.0.0104 online
Done
# lsdasd
0.0.(ECKD) at ( 94:  0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(ECKD) at ( 94:  4) is dasdb : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0300(FBA ) at ( 94:  8) is dasdc : active at blocksize 512, 524288 blocks, 256 MB
0.0.0301(FBA ) at ( 94: 12) is dasdd : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0102(ECKD) at ( 94: 16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0103(ECKD) at ( 94: 20) is dasdf : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at ( 94: 24) is dasdg : active at blocksize 4096, 600840 blocks, 2347 MB
```

Format the minidisk through the **dasdfmt** command and make a single partition through the **fdasd -a** command as you did before:

```
# dasdfmt -b 4096 -y -f /dev/dasdg
Finished formatting the device.
Rereading the partition table... ok
# fdasd -a /dev/dasdg
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

Creating a physical volume

Use the **pvcreate** command to create a physical volume from the minidisk:

```
# pvcreate /dev/dasdg1
```

```
  Physical volume "/dev/dasdg1" successfully created
```

Extending the volume group

Use the **vgextend** command to extend the volume group into the new physical volume. Then, use **vgdisplay** to verify that the volume group has free space.

```
# vgextend homevg /dev/dasdg1
  Volume group "homevg" successfully extended
# vgdisplay homevg
--- Volume group ---
  VG Name               homevg
  ...
  Total PE              1758
  Alloc PE / Size       1172 / 4.58 GB
  Free  PE / Size       586 / 2.29 GB
  VG UUID               XHdfZg-wHNg-Iqpm-lvzz-XGMW-JEMk-UBdgCU
```

Note there are 586 new free physical extents (PEs).

Extend the logical volume and the file system

Now that you have free space in the volume group, you can increase the size of the existing logical volume with the **lvextend** command. The **-l** option specifies the number extents to add. Finally, use the **resize2fs** command, which increases the size of the file system.

You can use the **df** command to show the file system size before and after you extend it, as the following example shows:

```
# df -h | grep home
/dev/mapper/homevg-homelv
                      4.6G  138M  4.2G   4% /home
# lvextend -l +586 /dev/homevg/homelv
  Extending logical volume homelv to 6.87 GB
  Logical volume homelv successfully resized
```

Attention: On some systems this can be done while the file system is mounted, but on SLES 10, it appears that it must be unmounted:

```
# resize2fs -p /dev/homevg/homelv
resize2fs 1.38 (30-Jun-2005)
/dev/homevg/homelv is mounted; can't resize a mounted filesystem!
```

Unmount the /home/ directory and first run **e2fsck** to verify the file system:

```
# umount /home
# e2fsck -f /dev/homevg/homelv
e2fsck 1.38 (30-Jun-2005)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/homevg/homelv: 35/600288 files (2.9% non-contiguous), 54018/1200128 blocks
```

Now use the **resize2fs** command to extend the logical volume, then remount /home/:

```
# resize2fs -p /dev/homevg/homelv
resize2fs 1.38 (30-Jun-2005)
Resizing the filesystem on /dev/homevg/homelv to 1800192 (4k) blocks.
```

```
Begin pass 1 (max = 18)
Extending the inode table     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/homevg/homelv is now 1800192 blocks long.

# mount /home
```

Use the **df -h** command to show that the file system is now 2.2 GB larger:

```
# df -h | grep home
/dev/mapper/homevg-homelv
                      6.8G  139M  6.4G   3% /home
```

Add minidisk 104 to the parameter line in /etc/zipl.conf and rerun the **zipl** command.

```
# cd /etc
# cp zipl.conf zipl.conf.works
# vi zipl.conf
# grep 104 zipl.conf
   parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-101,300-301,102-104
vmpoff=LOGOFF TERM=dumb"
# zipl
Using config file '/etc/zipl.conf'
...
```

You have now added the new minidisk as free space to the existing logical volume. It is recommended that you test your changes by rebooting your system:

```
# reboot
...
```

11.3 Centralizing home directories for LDAP users

If you have completed 10.2, “Creating a virtual LDAP server” on page 163, you have a working LDAP authentication server on LINUX02 that provides centralized login authentication. If you have completed 11.1, “Adding a logical volume” on page 180, you have a larger file system mounted over /home/ for storing user data.

Now you can bring these together with the automount service to have a centralized /home/ file system. In this fashion, users can login to all virtual servers using the same credentials and being able to access the same data. Further, you can modify the golden image so that all newly cloned virtual servers will inherit this setup.

Figure 11-2 on page 189 illustrates a block diagram.

Archived

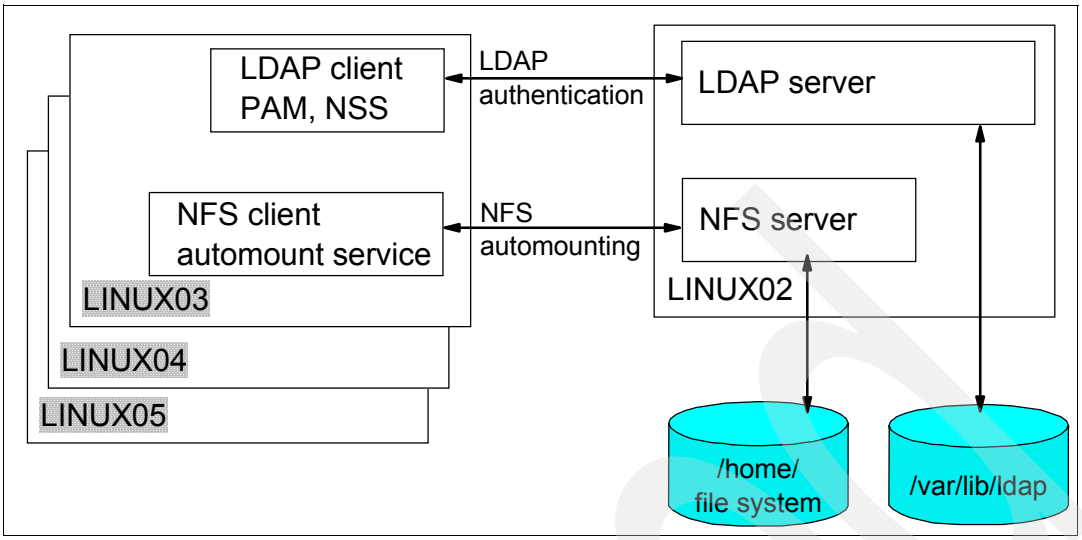


Figure 11-2 Block diagram of centralized LDAP authentication and automounted /home file system

11.3.1 Configuring the NFS server

Configure the Linux system on `LINUX02` to export `/home` over NFS. Add one line to the bottom of the `/etc/exports` file to export the `/home/` directory read/write:

```
# vi /etc/exports
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.
/home           *(rw,sync)
```

Set the NFS service to start at boot time through the **`chkconfig`** command and for this session through the **`service`** command:

```
# chkconfig nfsserver on
# service nfsserver start
Starting kernel based NFS server                           done
```

Test mount the newly exported directory locally:

```
# ls /mnt
# mount localhost:/home /mnt
# ls /mnt
lost+found  mikemac
# umount /mnt
```

This shows that the NFS server is set up and the `/home/` directory is available for mounting.

11.3.2 Configuring the golden image for LDAP and automount

In this section you modify the golden image. You will configure it to point the virtual server running on `LINUX02` for the following:

- LDAP authentication
- Automounting of `/home/` through NFS

Archive

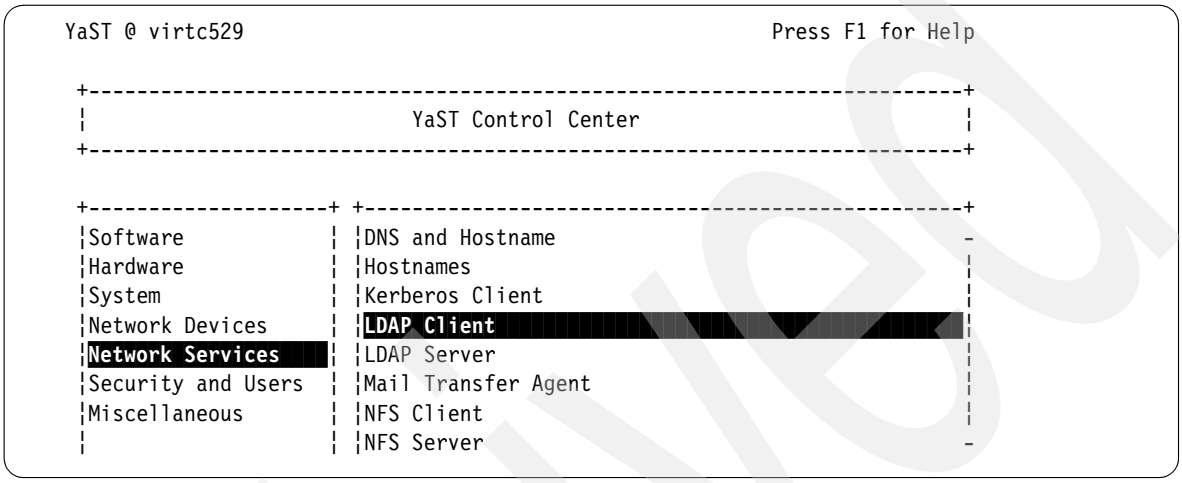
Configuring the golden image for LDAP authentication

To configure the golden image for LDAP authentication, first shut down the golden image. Then logon to SLES10S2. IPL Linux from 100, then start an SSH session as root to the golden image.

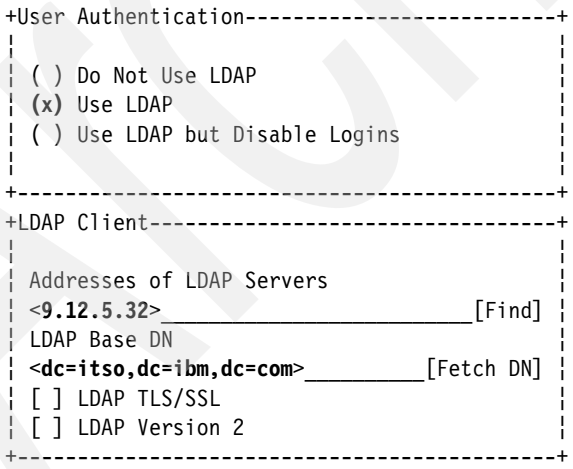
1. Again, **yast** can be used to configure the LDAP authentication client. Start **yast**:

```
# yast
```

2. Select **Network Services** on the left side of the main screen, then **LDAP Client**:



3. On the window that follows, use the **Tab** key to move to **Use LDAP** and press the space bar to select that choice.
4. Enter IP address of your LDAP server, which is 9.12.5.32 in this example.
5. Move to the **Fetch DN** button and press **Enter**. You should see that the OpenLDAP server is found.
6. Press **OK** in the result window. This way you can make sure LDAP server is accessible.



7. Use the Tab key to move to **Finish** and press **Enter**.
8. Some packages may need to be installed. Accept the default of **Continue**.
9. You will get a warning window; press **OK** there.
10. Press the **Quit** button on the main window to quit YaST.

Archived

11. Your LDAP client should now be pointing to the LDAP server. Test it through the **id** command to the ldap user:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=100(users),16(dialout),33(video)
```

This shows that the LDAP client is working.

Configuring automount

To configure automount, perform the following steps:

1. Install the automount package named `autofs` through the **yast -i** command:

```
# yast -i autofs
...
```

2. Add an entry to the bottom of `/etc/auto.master` for `/home`.

```
# cd /etc
# vi auto.master
#
# $Id: auto.master,v 1.4 2007/02/07 14:36:54 raven Exp $
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#/misc  /etc/auto.misc --timeout=60
#/smb   /etc/auto.smb
#/misc  /etc/auto.misc
#/net   /etc/auto.net
/home   /etc/auto.home
```

3. This line specifies that when the `/home/` directory is referenced, use the file `/etc/auto.home` to automatically mount it. Create the file `auto.home` and add one line to it:

```
# vi auto.home
*       9.12.5.32:/home/&
```

4. The asterisk (*) is used as a wildcard character to represent the local subdirectory. The ampersand (&) is used as the remote system name or the remote subdirectory. Whatever is in the local directory name field replaces the ampersand character.

5. Start **autofs** through the **service** command for this session and through **chkconfig** at boot time:

```
# service autofs start
Starting service automounter                                    done
# chkconfig autofs on
```

Testing autofs

This section describes how to test your changes.

1. Test the automounting of `/home/`. This can sometimes be problematic; if you **cd** to `/home/` and look for files, none may be listed.

```
# cd /home
# ls
```

2. No output from ls might lead you to believe that automount is not working. But if you **cd** into a directory to be automounted, then the mount will occur, as illustrated here:

```
# cd mikemac
# ls
.bash_history  .exrc     .mozilla  .xcoralrc          .xtalkrc
.bashrc        .fonts    .muttrc   .xemacs            Documents
```

```
.dvipsrc       .inputrc  .profile  .xim.template      bin
.emacs         .kermrc   .urlview  .xinitrc.template  public_html
```

3. Switch user to the LDAP user ID through the **su** command and create a test file:

```
# su - mikemac
mikemac@virtc529:~> echo "This is the file foo" > foo
mikemac@virtc529:~> exit
```

4. Then if you **cd** up one level, the directory remains automounted:

```
# cd ..
# ls
mikemac
```

This shows that autofs is working.

You have now configured the golden image, so autofs will mount the user's home directory under /home/.

Shut the system down so that it may be cloned.

```
# shutdown -h now

Broadcast message from root (pts/0) (Wed May 21 15:04:37 2008):

The system is going down for system halt NOW!
```

Cloning an update server

Go back to an SSH session on the controller.

1. Test cloning the updated golden image. In this example, the virtual server:

```
# clone.sh from sles10s2 to linux04
...
```

2. Start an SSH session to the newly cloned server as the LDAP user (not root):

```
login as: mikemac
Password: nottellin:)
```

3. Observe that the file created on the golden image is accessible on the new clone:

```
mikemac@virtc534:~> cat foo
This is the file foo
```

This shows that you have modified the golden image so that cloned virtual servers have a centralized LDAP server for authentication and a common home directory through NFS and automount.

11.4 Setting the software clock through NTP

It may be desirable to have all the system clocks on your Linux images be the same. This is extremely important for some applications. Synchronizing the clocks can be accomplished by setting up a time server on the controller, and having the virtual servers synchronize their times against the controller. The controller's time can be set accurately by having it synchronize against public time servers on the Internet. If you do *not* want your controller to communicate with the outside world, then do not set up the controller to synchronize with a public time server. If you *do* want to access an external public time server, then make sure your firewall allows traffic through between the controller and the public time servers on the well-known NTP port, 123.The standard open source time server on Linux is the Network

Time Protocol (NTP) daemon. The controller's NTP daemon needs to synchronize its time with public NTP servers.

There are two methods of utilizing public NTP servers on Internet. The traditional method was to choose random servers from a public list, and your system would be synchronized against only those servers. Now there is a new method of using a pool of NTP servers. This new method is recommended to avoid public server abuse. For background information about this topic, refer to the following Web site:

http://groups.google.com/group/comp.protocols.time.ntp/browse_frm/thread/fc51c6ee6ad66f66/9642ee4940b9982d#9642ee4940b9982d

Pool of NTP servers

DNS names of `0.pool.ntp.org`, `1.pool.ntp.org` and `2.pool.ntp.org` point to a set of random NTP servers throughout the world that change every hour. You can read more about this on the Web at:

http://www.pool.ntp.org/

To set up NTP, perform the following tasks:

1. Test to verify that the pool of servers is accessible by running **ntpdate**:

```
# ntpdate pool.ntp.org
5 Jan 09:47:24 ntpdate[7709]: step time server 89.212.6.198 offset -60.992462 sec
```

2. Modify the file `/etc/ntp.conf`. Comment out two lines that are pointing to an Undisciplined Local Clock and add two lines pointing to two external time references:

```
# cd /etc
# cp ntp.conf ntp.conf.orig
# vi ntp.conf   // change the line "server n.n.n.n" to point to controller
...
##
## Undisciplined Local Clock. This is a fake driver intended for backup
## and when no outside source of synchronized time is available.
##
#server 127.127.1.0            # local clock (LCL)
#fudge  127.127.1.0 stratum 10  # LCL is unsynchronized

##
## Outside source of synchronized time
##
## server xx.xx.xx.xx           # IP address of server
server 0.pool.ntp.org
server 1.pool.ntp.org
...
```

3. Start the ntp service for this session through the **service** command, and at boot time, through the **chkconfig** command:

```
# service ntp start
Try to get initial date and time via NTP from  0.pool.ntp.org 1.pool.doneorg
Starting network time protocol daemon (NTPD)                        done
# chkconfig ntp on
```

4. Issue the **ntptrace** command to see if the clock has synchronized. It takes some time for the software clock to synchronize:

```
# ntptrace
localhost: stratum 16, offset 0.000000, synch distance 0.006045
```

5. This shows that the software clock has not yet synchronized. The same command about about 20 minutes later gives a different output:

```
# ntptrace
localhost: stratum 3, offset 0.008162, synch distance 0.093268
...
```

This shows that the software clock has synchronized to become a stratum 3 time source.

11.5 Rescuing a Linux system

This section describes how to boot your Linux server into different modes for troubleshooting purposes. It covers booting Linux into single user mode, and also entering a rescue environment when you require more advanced troubleshooting.

11.5.1 Entering single user mode

Single user mode is helpful when you need to recover the root password, or if you are having problems while booting Linux into the default run level. To enter single user mode, first IPL your Linux server from the 3270 console. Enter **#cp vi vmsg 0 1** boot the desired menu option (0), followed by the run level (1 for single user mode). If you have turned the default timeout back to 3 seconds from 10 seconds, you have to type quickly:

```
00: zIPL v1.6.3 interactive boot menu
00:
00:  0. default (ipl)
00:
00:  1. ipl
00:  2. Failsafe
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 3 seconds):
00:
==> #cp vi vmsg 0 1
00: CP VI VMSG 0 1
00: Booting default (ipl)...
Linux version 2.6.16.60-0.21-default (geeko@buildhost) (gcc version 4.1.2 200701
15 (SUSE Linux)) #1 SMP Tue May 6 12:41:02 UTC 2008
We are running under VM (64 bit mode)
Detected 1 CPU's
Boot cpu address  0
Built 1 zonelists
```

Note that the number 1 is appended to the kernel parameters:

```
Kernel command line:
root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-101,300-301 vmpoff=LOGOFF TERM=dumb 1
BOOT_IMAGE=0
...
```

Many messages will pass by, then the Linux system will enter run level 1:

```
...
Ÿ1A..done INIT: Entering runlevel: 1
 Boot logging started on /dev/ttyS0(/dev/console) at Thu May 22 06:56:31 2008
Master Resource Control: previous runlevel: N, switching to runlevel: 1
Sending all processes the TERM signal...
```

```
..done
Sending all processes the KILL signal...
..done
 INIT: Going single user
 INIT: Sending processes the TERM signal
Give root password for login:
lnx4vm
virtc531:~ # runlevel
1 S
```

In single user mode, you are logged in as the root user. You can use the **passwd** command to set the root password. All of the file systems in `/etc/fstab` are mounted, but networking has not been started. To exit single user mode, you can type `reboot`, or enter `init 3` to continue booting normally.

11.5.2 Entering a rescue environment

If you encounter errors mounting the root file system, or have other problems that prevent you from entering single user mode, you can enter a rescue environment. To enter into rescue environment you start installation of Linux again, but will not go further than activating DASDs.

To enter a rescue environment, initiate an interactive Linux installation. Because you will have to SSH into installation image during the installation, you cannot use VNC because the SSH server is not started in a VNC type of install. You will have to use SSH. Perform the following steps:

1. Logon to `LNXMAINT`.
2. Copy the user's `PARM-S10` file to a new file (`LINUX04 RESCUE`, in this example). Edit the `LINUX04 RESCUE` file, changing the `UseVNC` and `VNCpassword` variables to the **UseSSH** and **SSHPassword** variables:

```
==> copy linux04 parm-s10 d = rescue =
==> x linux04 rescue d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.5.34 Hostname=virtc534.itso.ibm.com
Gateway=9.12.4.1 Netmask=255.255.252.0
Broadcast=9.12.4.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7 Portname=dontcare
Install=nfs://9.12.4.185/install/sles10sp2/SLES-10-SP2-DVD-s390x-GMC-DVD1.iso
UseSSH=1  SSHPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

3. Copy the `SLES10S2 EXEC` file to a new file named `RESCUE EXEC`. Edit the new file to point to the new `RESCUE` parameter file:

```
==> copy sles10s2 exec d rescue = =
==> x rescue exec
...
00005 'PUN SLES10       KERNEL * (NOH'
00006 'PUN' userid() '  RESCUE * (NOH'
00007 'PUN SLES10       INITRD * (NOH'
...
```

4. Log off from `LNXMAINT` and logon to `LINUX04`.
5. Respond `no` to the IPL question, and run the `RESCUE EXEC`.

```
==> rescue
00:      NO FILES PURGED
```

```
00: RDR FILE 0001 SENT FROM LINUX04  PUN WAS 0001 RECS 089K CPY  001 A NOHOLD NOKEEP
00: RDR FILE 0002 SENT FROM LINUX04  PUN WAS 0002 RECS 0010 CPY  001 A NOHOLD NOKEEP
00: RDR FILE 0003 SENT FROM LINUX04  PUN WAS 0003 RECS 107K CPY  001 A NOHOLD NOKEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Linux version 2.6.16.60-0.17-default (geeko@buildhost) (gcc version 4.1.2 200701
15 (SUSE Linux)) #1 SMP Tue Apr 22 07:28:35 UTC 2008
We are running under VM (64 bit mode)
Detected 1 CPU's
Boot cpu address  0
Built 1 zonelists
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dum
b                    HostIP=9.12.5.34 Hostname=virtc534.itso.ibm.com
                     Gateway=9.12.4.1 Netmask=255.255.252.0
                     Broadcast=9.12.4.255 Layer2=0
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
0.0602               Nameserver=9.12.6.7 Portname=dontcare
                     Install=nfs://9.12.5.30/nfs/sles10sp2/SLES-10-SP2-DVD-s390x
-GMC-DVD1.iso        UseSSH=1  SSHPassword=lnx4vm
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
...
```

6. The install process directs you to SSH to the IP address of your Linux server to begin the first stage of the installation:

```
***  sshd has been started  ***


      ***  login using 'ssh -X root@virtc534.itso.ibm.com'  ***
      ***  run 'yast' to start the installation  ***
```

7. Start an SSH session to the rescue system as root.

```
login as: root
root@9.12.5.34's password:

SUSE Linux Enterprise Server 10 Installation

Welcome to the inst-sys...
Linux virtc534.itso.ibm.com 2.6.16.60-0.21-default #1 SMP Tue Apr 22 07:28:35 UTC 2008
s390x s390x s390x GNU/Linux
/root

run yast to start the installation

inst-sys:~ #
```

8. Run **yast**. After choosing the language, accept the licence agreement and then choose Configure DASD Disks and Activate your disks.

9. Start another SSH session. You should be able to mount the root file system:

```
inst-sys:~ # mount /dev/dasda1 /mnt
inst-sys:~ # ls /mnt
bin   dev  home  lib64       media  opt   root  srv  tmp  var
boot  etc  lib   lost+found  mnt    proc  sbin  sys  usr
```

From this rescue environment, you should be able to modify the system as needed.

Archived

11.5.3 Integrated ASCII console

When using a 3270 terminal for accessing Linux, editing Linux files is a cumbersome task. A rescue can be done through the Integrated ASCII console easily, because it provides a real VT220 terminal emulator which allows you to edit Linux files in vi, for example. Using the Integrated ASCII console requires you to performs some tasks in advance.

1. Add console statements into the `/etc/zipl.conf` file:

```
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 dasd=100-101,300-301
vmpoff=LOGOFF TERM=dumb console=ttyS0 console=ttyS1"
```

2. Run **zipl**.
3. Edit the console definition in `/etc/inittab`, then uncomment the second statement in the console definition and change `xterm` to `vt220`:

   Before:

```
# Default HMC/3215/3270 console:
1:2345:respawn:/sbin/mingetty --noclear /dev/ttyS0 dumb
# VT220(Linux) console:
#2:2345:respawn:/sbin/mingetty --noclear /dev/ttyS1 xterm
# additional 3270 terminals:
#3:2345:respawn:/sbin/mingetty --noclear /dev/3270/ttycons dumb
```

   After:

```
# Default HMC/3215/3270 console:
1:2345:respawn:/sbin/mingetty --noclear /dev/ttyS0 dumb
# VT220(Linux) console:
2:2345:respawn:/sbin/mingetty --noclear /dev/ttyS1 vt220
# additional 3270 terminals:
#3:2345:respawn:/sbin/mingetty --noclear /dev/3270/ttycons dumb
```

4. Check `/etc/securetty`; it should contain ttyS1.
5. Reboot Linux.

Now when you want to use the Integrated ASCII console, run it from HMC (**Recovery** -> **Integrated ASCII console**) and attach the console to the Linux image from the z/VM session:

```
ATTACH SYSASCII TO <linux01>
```

If you need to attach it to another image, first detach it:

```
DET SYSASCII <linux01>
```

11.6 Configuring rsync

The **rsync** command copies just the differences or deltas of changed files. For this reason, it is an efficient copy tool. Following is an example of using rsync to back up the `/etc/` file system of another virtual server. There are two files that need to be configured: the main configuration file `/etc/rsyncd.conf`, and a file on the target system to allow only the controller to connect, `/etc/trusted.hosts`.

Perform the following steps:

1. Use the **chkconfig** command to set **rsync** to start up:

```
# chkconfig rsync on
```

Archived

2. First make a backup cop, then edit the file /etc/rsyncd.conf, commenting out two lines and adding five lines at the bottom.

```
# cd /etc
# cp rsyncd.conf rsyncd.conf.orig
# vi rsyncd.conf
gid = users
read only = true
# use chroot = true
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
hosts allow = trusted.hosts
# slp refresh = 300
max connections = 1

[etc]
path = /etc
read only = yes
list = yes
```

3. The new [etc] section at the bottom will allow access from the controller to the /etc/ directory.
4. The file /etc/trusted.hosts needs a single entry, which is the TCP/IP address of the virtual server that will be backing it up (9.12.5.30, in this example). This will allow only that server to get an **rsync** session. Create this file:

```
# vi trusted.hosts
<9.12.5.30>
```

11.7 Enabling VMRM to work with CMM

VMRM is a z/VM tool that provides functions to dynamically tune the z/VM system. Groups of virtual machines can be defined to be part of a workload. The workloads can then be managed by VMRM to goals that are also defined.

Cooperative Memory Management (CMM1) is a facility which provides z/VM with a way to indirectly control the amount of memory that Linux uses. This technique is known as “ballooning”, and is implemented by a special driver in Linux. When VMRM determines that the system is becoming too constrained by memory, it will direct the driver to allocate and pin memory within Linux, which forces Linux to relinquish or page out some of the memory it had been using. In turn, the driver informs z/VM of the pages it has pinned. Because the driver was designed never to use these pages, z/VM knows they are now available for it to use, and so can reclaim them, thus reducing overall system memory pressure.

On the other hand, when VMRM determines that the system is no longer under memory pressure, it will direct the ballooning driver to reduce the number of pages it has pinned, thus allowing Linux to expand its memory usage once more.

More details about CMM1 can be found at:

http://www.vm.ibm.com/sysman/vmrm/vmrmcmm.html

Collaborative Memory Management Assist

Collaborative Memory Management Assist (CMMA) consists of changes in hardware, z/VM 5.3, and Linux. The hardware facility includes a new instruction (EXTRACT AND SET STORAGE

ATTRIBUTES - ESSA), which can be simulated by z/VM on machines where the facility is not available (although, due to simulation overhead, it is generally not recommended to do so).

CMMA introduces new usage and state information for each memory page, thus enabling Linux and z/VM to employ more efficient memory management algorithms which do not conflict with one another.

More details about CMMA can be found in *z/VM V5R3.0 CP Programming Services*, SC24-6084, at:

http://publibz.boulder.ibm.com/epubs/pdf/hcse5b21.pdf

In both CMM flavors, the benefit is proportional to the potential. On systems where the guests are poorly configured, or over-sized, you will see the most benefit. There was an early study done by a group in IBM, “z/VM Large Memory - Linux on System z”, available at:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101151

CMMA is not described in detail, but it is mentioned for completeness.

11.7.1 Configuring CMM1

CMM1 must be enabled on both the Linux and z/VM sides for it to function. To enable it for Linux, see 8.5.4, “Enabling the cmm module” on page 136.

To enable z/VM for CMM1, perform the following steps:

1. Log on to the user ID VMRMSVM.
2. Create a new configuration file, or edit the existing configuration file VMRM CONFIG A. Add two lines as follows:

```
==> x vmrm config
ADMIN MSGUSER VMRMADMN
NOTIFY MEMORY LINUX* SLES10* RHEL5*
```

   This example will apply to z/VM user IDs whose names begin with LINUX, SLES10, or RHEL5.

3. Edit the PROFILE EXEC on AUTOLOG1 so that VMRMSVM is autologged at IPL time. Logon to AUTOLOG1.
4. Before pressing Enter at the VM READ prompt, type acc (noprof so that the PROFILE EXEC is not run. Then add:

```
LOGON AUTOLOG1
z/VM Version 5 Release 4.0, Service Level 0801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 09:07:43 EDT SUNDAY 05/25/08
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
==> acc (noprof
==> x profile exec
/**************************/
/* Autolog1 Profile Exec   */
/**************************/
'cp xautolog tcpip'          /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp xautolog vmrmsvw'        /* start VMRM service virtual machine */
```

```
'cp set pf12 ret'            /* set the retrieve key */
...
```

5. Start the user ID for this IPL of z/VM:

```
==> xautolog vmrmsvm
Command accepted
AUTO LOGON  ***       VMRMSVM  USERS = 29
Ready; T=0.01/0.01 14:08:09
HCPCLS6056I XAUTOLOG information for VMRMSVM: The IPL command is verified by the
 IPL command processor.
```

11.7.2 How to observe that CMM1 is working

To determine whether CMM1 is working:

- From the z/VM perspective, you should see resident pages and total pages for the CMM1-managed virtual machines drop when the system comes under constraint.

  For Performance Toolkit, this would be seen in UPAGE report (FCX113).

- The ballooning driver releases the pages it has pinned by using Diagnose x'10'. You can see the diagnose 10 counts in PRIVOP report (FCX104).
- Other items of interest may be users in page wait; see USTAT (FCX114).
- From within Linux, you can check the size of the pool of pages put out of play through the commands:

```
# cat /proc/sys/vm/cmm_pages
# cat /proc/sys/vm/cmm_timed_pages
```

Archived

12

Monitoring z/VM and Linux

This chapter briefly describes how to monitor z/VM and Linux. For more detailed information about z/VM performance and monitoring, see Chapter 11, “Monitoring performance and capacity”, in the *Getting Started With Linux*, SC24-6096, which is available on the Web at:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/hcsh2a70

There are a number of z/VM monitoring tools such as the CA VM:Monitor, the IBM z/VM Performance Toolkit, and the IBM Tivoli OMEGAMON® XE for z/VM and Linux. The IBM z/VM Performance Toolkit is briefly described in this section.

For more information about the CA VM:Monitor, see:

http://www.ca.com/

This chapter describes the following topics:

- Using INDICATE and other commands
- The z/VM Performance Toolkit
- Monitoring Linux
- Viewing Linux data in the Performance Toolkit

© Copyright IBM Corp. 2008. All rights reserved.

12.1 Using INDICATE and other commands

z/VM has many commands to monitor the state of the system. **CP INDICATE** is the most commonly used, and there are other commands that are addressed.

12.1.1 Using the INDICATE command

z/VM has basic commands such as **INDICATE**. This command can be useful if there are no other tools with more extended function present on the system, such as when undertaking a completely new system installation. (However, command level tools should not be relied upon for formal, long-term reporting on production systems.)

There are many **INDICATE** parameters that can be included as command line options. Use the command **HELP INDICATE** for a basic understanding, and then press **F11** for help on each parameter.

INDICATE LOAD

If no parameter is specified, **INDICATE LOAD** is the default option. There are two “flavors” of this, depending on whether the issuing user ID has privilege class G or privilege class E. Class G users can use **INDICATE** to display recent contention for system resources, as well as to display environment characteristics and measurements of resources used by their virtual machine.

The output from user ID with class E privilege (for example, MAINT, OPERATOR) is shown here. The lines are numbered for reference to the descriptions that follow.

```
==> ind load
1  AVGPROC-038% 03
2  XSTORE-000021/SEC MIGRATE-0001/SEC
3  MDC READS-000068/SEC WRITES-000001/SEC HIT RATIO-099%
4  PAGING-0031/SEC STEAL-000%
5  Q0-00006(00000)                              DORMANT-00357
6  Q1-00001(00000)           E1-00000(00000)
7  Q2-00001(00000) EXPAN-002 E2-00000(00000)
8  Q3-00034(00000) EXPAN-002 E3-00000(00000)
9
10 PROC 0000-038%          PROC 0001-038%
11 PROC 0002-038%
12
13 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 you see here are a smoothed average over the past 4 minutes. Areas where z/VM performance analysts tend to focus are the following:

1 AVGPROC gives the overall processor utilization (38%, in this example). The number following it is the number of online processors (3, in this example). The individual processor utilization is shown on lines 10 and 11. Examine these to see if they are somewhat balanced. There are cases where an imbalance is acceptable. This would include very low utilization scenarios, or cases where there are not enough users ready to run virtual processors to keep the physical processors busy. One of the processors will be a Master, all of the others Alternate, and some imbalance may result from performing these functions.

2 Paging to expanded storage is displayed here. Most z/VM systems on z9 class machines can sustain several thousands of this type of paging operations a second without any

problems. The MIGRATE rate is the number of pages per second being moved from expanded storage out to paging space on DASD. A healthy system will have a MIGRATE rate significantly lower than the XSTORE rate, probably being measures in hundreds rather than thousands. The higher values seen tend to build up over time, and are sustained over periods of intense system activity; however, there are times when the MIGRATE value may spike for brief periods of time.

3 Minidisk cache (MDC) statistics are displayed here. The effectiveness of MDC can be judged by the combination of the READS rate and the HIT RATIO. If both are high, then a large number of physical I/Os are avoided due to the MDC feature. For a system that has an appreciably high I/O rate, composed of reads plus writes, and a high proportion of reads, and a good hit ratio for those reads (tending to 90% or greater), the real, physical I/O avoidance can be very high (as high as 50% in some cases has been observed). Conversely, however, a high HIT RATIO with a low value for the READS rate should not be taken as desirable (a 100% hit ratio, when doing only one I/O per second, is effectively meaningless).

4 More storage (memory) management is displayed here. The PAGING rate is important. Higher values will often impact performance. This 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.The STEAL percentage is often misleading. This is basically the percentage of pages taken from guests that z/VM believes are non-dormant. Because some guests have periodic timers going off, they appear to be active to z/VM even when relatively idle. Pages taken from these guests are still considered to be stolen. So there are scenarios where a system only has a user set comprising active guests, in which case all pages taken would be considered stolen. Bearing this in mind, if a high STEAL value is observed, the paging rate needs to be checked. If the paging rate is relatively low, then the STEAL value is not important.

On lines 5 through 8, you also see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into three different classes (1 through 3), and a special additional class labelled zero. So the Column of Qx values and Ex represent the virtual machines in the dispatch list and the eligible list. The most important value here to validate is that there are no virtual machines in the Eligible list: E1, E2, E3; this implies z/VM has stopped dispatching some virtual machines to avoid over-committing resources. Such a system would require further investigation, possibly leading to some tuning work, or even the addition of some hardware in extreme cases. You can ignore the values in parenthesis.

INDICATE QUEUES EXP

Another useful command to help you determine the state of the system is **INDICATE QUEUES EXP**. Following is an example:

```
==> ind q exp
DATAMGT1      Q3 AP  00000537/00000537 .... -2.025 A02
BITNER        Q1 R00 00000785/00000796 .I.. -1.782 A00
EDLLNX4       Q3 PS  00007635/00007635 .... -1.121 A00
TCPIP         Q0 R01 00004016/00003336 .I.. -.9324 A01
APCTEST1      Q2 IO  00003556/00003512 .I.. -.7847 A01
EDLWRK20      Q3 AP  00001495/00001462 .... -.6996 A01
EDL           Q3 IO  00000918/00000902 .... -.2409 A01
EDLWRK11      Q3 AP  00002323/00002299 .... -.0183 A00
EDLWRK18      Q3 IO  00001052/00000388 .... -.0047 A00
EDLWRK4       Q3 AP  00004792/00002295 ....  .0055 A01
EDLWRK8       Q3 AP  00004804/00004797 ....  .0089 A02
EDLWRK16      Q3 AP  00002378/00002378 ....  .0170 A02
EDLWRK2       Q3 AP  00005544/00002956 ....  .0360 A00
EDLWRK12      Q3 AP  00004963/00002348 ....  .0677 A01
EDLWRK6       Q3 IO  00000750/00000302 ....  .0969 A02
EDLWRK3       Q3 AP  00005098/00005096 ....  .0999 A02
EDLWRK17      Q3 AP  00004786/00004766 ....  .1061 A01
```

```
EDLWRK9       Q3 AP  00002372/00002334 ....  .1107 A02
EDLWRK5       Q3 IO  00002376/00002376 ....  .1205 A01
EDLWRK14      Q3 AP  00002426/00002323 ....  .1238 A02
EDLLIB19      Q3 IO  00001226/00001100 ....  .1309 A02
EDLWRK19      Q3 AP  00002322/00002298 ....  .1705 A00
EDLWRK15      Q3 AP  00002839/00002781 ....  .2205 A02
EDLWRK1       Q3 AP  00002969/00002935 ....  .2491 A02
```

This is another class E command, and it displays the virtual processors associated with a given user ID (a single virtual machine may have multiple virtual processors), along with what queue (dispatch list, eligible list, limit list) they are in and what state they are in. This is a snapshot in time.

Check this output to make sure there are no virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Q_x (x=1,2,3). Eligible list would be marked as E_x.

The third column in the example displays the state of the virtual processor. This information can give you an idea of how the virtual processors might be constrained. Virtual processors that are actually running in the snapshot period are marked with RNN, where NN is the processor number they are on. An `R` without a number means the virtual processor is ready to run, but there is no processor available. (Note that the virtual machine that issues the **`INDICATE`** command will always be one of the running machines.) Other states are documented in the help information for the **`IND Q EXP`** command.

You can ignore the remaining columns unless detailed analysis is required, or IBM support requests it. Also, always remember that is just a snapshot in time and repeating this command over time can often provide a more accurate picture of your z/VM system; a single snapshot cannot be regarded as indicative.

12.1.2 Using other basic commands

This section briefly mentions other basic commands that are especially useful. All examples are shown from the `MAINT` user ID. The results will be different for users with fewer privileges.

Getting help

To get help on the system, use the **`HELP`** command. Here are some useful help commands:

```
==> 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
LINUX06  - DSC , LINUX04  - DSC , LINUX03  - DSC , LINUX07  - DSC
LINUX01  - DSC , SLES9    - DSC , FTPSERVE - DSC , DTCVSW2  - DSC
DTCVSW1  - DSC , TCPIP    - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP     - DSC , OPERATOR - DSC , MAINT    -L0005
VSM     - TCPIP
```

Determining storage or memory

To see how much central and expanded storage (memory) are installed and allocated to a system, use the **`QUERY STORAGE`** and **`QUERY XSTOR`** commands. For example:

```
==> q stor
```

```
STORAGE = 4G CONFIGURED = 4G INC = 256M STANDBY = 0  RESERVED = 0
==> q xstor
XSTORE= 2048M online= 2048M
XSTORE= 2048M userid= SYSTEM usage= 0% retained= 0M pending= 0M
XSTORE MDC min=0M, max=0M, usage=0%
XSTORE= 2048M userid=  (none)  max. attach= 2048M
```

Determining processors or CPUs

To see how many processors (CPs, IFLs, CPUs) you have 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 STANDBY CP
PROCESSOR 05 STANDBY CP
PROCESSOR 06 STANDBY CP
PROCESSOR 07 STANDBY CP
```

Determining software level

To determine what level of CP your system is at, use the **QUERY CPLEVEL** command. For example:

```
==> q cplevel
z/VM Version 5 Release 4.0, service level 0801 (64-bit)
Generated at 04/08/08 18:17:39 EDT
IPL at 05/18/08 10:17:32 EDT
```

Determining system cylinder allocation

The **QUERY ALLOC MAP** command shows you the system allocation of spool, paging and directory space. For example:

```
==> q alloc map
EXTENT    EXTENT                        % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
MVD850 D850          1         20     20      1      2   5% DRCT ACTIVE
MVD851 D851          1       3338 600840  58332  63360   9% SPOOL
MVD852 D852          1       3338 600840      0      0   0% PAGE
MPD855 D855          0       3338 601020      0      0   0% PAGE
MPD950 D950          0       3338 601020      0      0   0% PAGE
MPD951 D951          0       3338 601020      0      0   0% PAGE
MPDA50 DA50          0       3338 601020      0      0   0% PAGE
MPDB50 DB50          0       3338 601020     13     45   1% PAGE
```

Determining DASD, OSA and virtual resources

The **QUERY DASD** and **QUERY DASD FREE** commands will show you what DASD is assigned to the system and what DASD is free to be assigned. Similarly, the **QUERY OSA** and **QUERY OSA FREE** commands report on OSA resources. Finally, the **QUERY VIRTUAL ALL** command can be useful.

The following list gives the short form of these commands, without associated output:

```
==> q da
==> q da free
==> q osa
==> q osa free
==> q v all
```

Archived

12.2 z/VM Performance Toolkit

To use the z/VM Performance Toolkit, the product must be licensed. Only configure the product if you have licensed it.

For more detailed information about the toolkit, refer to the following documentation:

- *z/VM Performance Toolkit Guide*, SC24-6156, and *z/VM Performance Toolkit Reference*, SC24-6157, which are available on the Web starting at the z/VM 5.4 bookshelf:

  http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm54

  Search for **Toolkit** on that page.
- *The Program Directory for Performance Toolkit for VM*, GI10-0785-00

  http://www.vm.ibm.com/progdir/5vmptk30.pdf
- IBM Redbooks publication *Linux on IBM zSeries and S/390®: Performance Toolkit for VM*, SG24-6059

  http://www.redbooks.ibm.com/abstracts/sg246059.html

The following sections provide a brief explanation of how to set up and use the IBM Performance Toolkit.

12.2.1 Configuring the z/VM Performance Toolkit

The Performance Toolkit is installed with z/VM. Configuration is described in the Program Directory. Following is a summary of how to turn it on. As mentioned, only configure the product if you have licensed it.

1. Query which priced products are enabled through the **QUERY PRODUCT** command:

```
==> q product
Product  State    Description
5VMDIR40 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FL 540
5VMPTK40 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
5VMRAC40 Disabled 00/00/00.00:00:00.$BASEDDR RACF for VM
5VMRSC40 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking Version 5 Release 4
 Modification 0
```

2. To enable The z/VM Performance Toolkit, logon to MAINT 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 should see a few screens of messages scroll by and finally the success messages shown above. This will enable the Performance Toolkit for the current z/VM session.

3. At IPL time, the SYSTEM CONFIG file is modified by having a line appended to the end. Verify this with the following commands:

```
==> acc cf1 f
==> x system config f
====> bot
====> -2
====> pre off
...
```

```
PRODUCT PRODID 5VMPTK40 STATE ENABLED DESCRIPTION '05/22/08.10:08:55.MAINT    PE
RFKIT Minidisk Install and Service'
```

The Performance Toolkit is now enabled.

12.2.2 Configuring Web Browser support

After the product is enabled, the TCPIP profile must be modified to add browser capabilities for the Performance Toolkit. The following example describes how to set the port to 80, which is the default for a Web browser.

1. Logon to TCPMAINT. Edit the <vmlinuxa> TCPIP D file (assuming you modified this file name earlier; the default name is PROFILE TCPIP) and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved.

   ```
   ==> x <vmlinuxa> tcpip d
   ====> /port
   ```

2. Add the following lines 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
   ; 53   TCP NAMESRV             ; Domain Name Server
   ; 53   UDP NAMESRV             ; Domain Name Server
   ; 67   UDP BOOTPD              ; BootP Server
   ; 67   UDP DHCPD               ; DHCP Server
   ; 69   UDP TFTPD               ; TFTPD (Trivial FTP) Server
     80   TCP PERFSVM            ; Performance Toolkit
   ; 111  TCP PORTMAP             ; Portmap Server
   ...
   ```

   Save your changes. The TCPIP user ID needs to be recycled in order for your changes to take effect. You can issue **FORCE** and **XAUTOLOG TCPIP** from a console. Alternatively, if you are in a position to reIPL the system, you can do that (shutdown reipl iplparms cons=sysc).

3. When the system comes back, logon to TCPMAINT and verify success by issuing the **NETSTAT** command. Verify that the service PERFSVM is in a Listen state.

   ```
   ==> netstat
   VM TCP/IP Netstat Level 540       TCP/IP Server Name: TCPIP

   Active IPv4 Transmission Blocks:

   User Id  Conn    Local Socket            Foreign Socket          State
   ---- --  ----    ----- ------            ------- ------          -----
   FTPSERVE 1001    *..FTP-C                *..*                    Listen
   INTCLIEN 1009    *..TELNET               *..*                    Listen
   PERFSVM  1011    *..80                   *..*                    Listen

   Active IPv6 Transmission Blocks: None
   ```

   In this case, the PERFSVM service was shown to be listening on port 80.

Archived

12.2.3 Configuring PERFSVM

The PERFSVM user ID is the Performance Toolkit service machine.

1. Logon to PERFSVM. If you have successfully enabled the product, you should be put into a Performance Toolkit session and should see the following text at the top of the window:

```
FCX001                Performance Toolkit for VM
FCXBAS500I Performance Toolkit for VM FL530 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

(You may also receive the message FCXPMN446E Incomplete monitor data: SAMPLE CONFIG size too small. For an explanation of how to resolve this issue, refer to 12.5, “Common Performance Toolkit issue” on page 217.)

2. Press **PF12** twice to get to a CMS prompt.
3. Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.
   a. Use the **VMLINK** command to both link the disk read-only and access it as the highest available file mode. The default read password is **read**; however, if you changed your passwords as described in 4.9.3, “Changing passwords in USER DIRECT” on page 61, then the password will be lnx4vm.

```
==> vmlink maint 191
ENTER READ PASSWORD:
lnx4vm
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

   b. Copy the PROFILE XEDIT to the A disk:

```
==> copy profile xedit z = = a
```

4. Copy the default configuration files, which are on the PERFSVM D disk, to your A disk:

```
==> copy * * d = = a
```

5. The main configuration file is FCONX $PROFILE. Edit that file and search for the string VMCF. This should take you to line 173, where the next four lines are comments starting with an asterisk (*). Make the following changes:
   – Uncomment the second line and the fourth line by changing *C to **FC**.
   – Change port 81 to 80 on the fourth line.
   – Add the text FC MONCOLL LINUXUSR ON TCPIP TCPIP after the fourth line. This tells the Performance Toolkit to collect Linux performance data.

   The modified lines should look as follows. Save your changes with the **FILE** subcommand:

```
==> x fconx $profile a
====> /vmcf
*    Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
*    Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 80
FC MONCOLL LINUXUSR ON TCPIP TCPIP
*    Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
====> file
```

6. Create a remote data retrieval authorization file with your z/VM system identifier; replace <vmlinuxa> with your z/VM system name:

```
==> x fconrmt authoriz
```

```
====> a 2
<vmlinuxa>  PERFSVM  S&FSERV
<vmlinuxa>  MAINT    DATA CMD EXCPMSG
```

7. Create a system identification file; replace <vmlinuxa> with your z/VM system name:

```
==> x fconrmt systems
====> a
<vmlinuxa>   PERFSVM  z/VM5.4 N   FCXRES00
```

8. Create a Linux system definition file. Add the TCP/IP addresses of your Linux system(s). The following example shows adding five Linux virtual servers:

   Use port 8803 for Linux performance data:

```
==> x fconx linuxusr a
LINUX00 <9.12.5.30>:8803
LINUX01 <9.12.5.31>:8803
LINUX02 <9.12.5.32>:8803
LINUX03 <9.12.5.33>:8803
LINUX04 <9.12.5.34>:8803
```

9. Edit the PROFILE EXEC file and uncomment the five MONITOR SAMPLE statements and the two MONITOR EVENT statements. Also, the size of the of the SAMPLE CONFIG and EVENT CONFIG may need to be increased:

```
==> x profile exec a
```

   Before:

```
...
/*** Once you have PERFKIT enabled and running uncomment the      ***/
/*** following comments                                           ***/
/* 'CP MONITOR SAMPLE ENABLE PROCESSOR'    */
/* '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 EVENT  ENABLE STORAGE'      */
/* 'CP MONITOR EVENT  ENABLE I/O ALL'      */

'PERFKIT'                   /* Invoke the PERFKIT module   @FC012BD*/

Exit
```

   After:

```
...
/*** Once you have PERFKIT enabled and running uncomment the      ***/
/*** following comments                                           ***/
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'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 EVENT  ENABLE STORAGE'
'CP MONITOR EVENT  ENABLE I/O ALL'

/* Increase the size of the SAMPLE CONFIG and EVENT CONFIG */
'CP MONITOR SAMPLE CONFIG SIZE 1200'
'CP MONITOR EVENT CONFIG SIZE 350'

'PERFKIT'                   /* Invoke the PERFKIT module   @FC012BD*/
```

Archived

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

You should now be ready to run the Performance Toolkit.

12.2.4 Starting the z/VM Performance Toolkit

To start the Performance Toolkit, enter the following command from the PERFSVM user ID:

```
===> perfkit
FCX001                Performance Toolkit for VM              Autoscroll 12
FCXBAS500I Performance Toolkit for VM FL540 25Feb08
FCXAPP530I Connected to *IDENT for resource FCXRES00
FCXAPF530I Connected to *IDENT for resource FCXSYSTM
FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
FCXTCR571I Connected to TCP/IP server TCPIP on path 0004
FCXAPP527I User PERFSVM connected on path 0007
FCXAPC535I Connected to resource FCXRES00 on path 0006, for S&F-Coll
FCXTCP575I WebServer host IP address is 9.12.5.22:00080
FCXTCR575I TCPRequest host IP address is 9.12.5.22:41780
FCXTCR590I TCPRequest interface activated
FCXTCP590I WebServer interface activated
```

Disconnect from PERFSVM now.

```
Command ===> disc
```

The Performance Toolkit should now be configured and running. It is now collecting basic data. However, to get relevant data from Linux images, they must be configured appropriately as described later.

12.2.5 Using the z/VM Performance Toolkit

The Performance Toolkit can be used through 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 at port 80. For example:

   ```
   http://9.12.5.22
   ```

2. You should see your system on the Web Session Setup window. Click it and you will be presented with the Web Server Logon window; see Figure 12-1 on page 211.

Archived

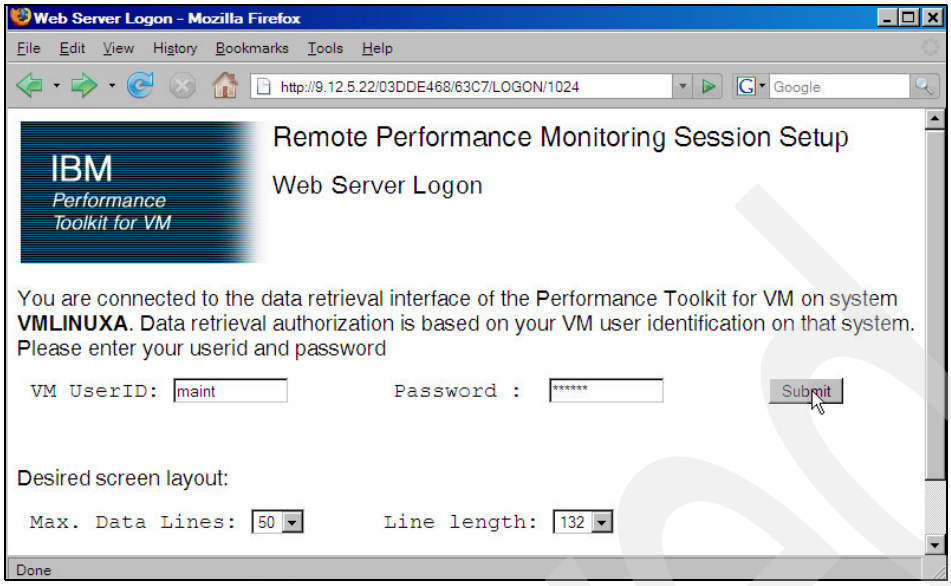


Figure 12-1 Performance Toolkit logon window

3. Enter any valid user ID and password (in this example, `MAINT`).
4. You should see the Central Monitoring System Load Overview with your system name on the left side.
5. Click your system name and you should see the Initial Performance Data Selection Menu window, as shown in Figure 12-2 on page 212.

Archived

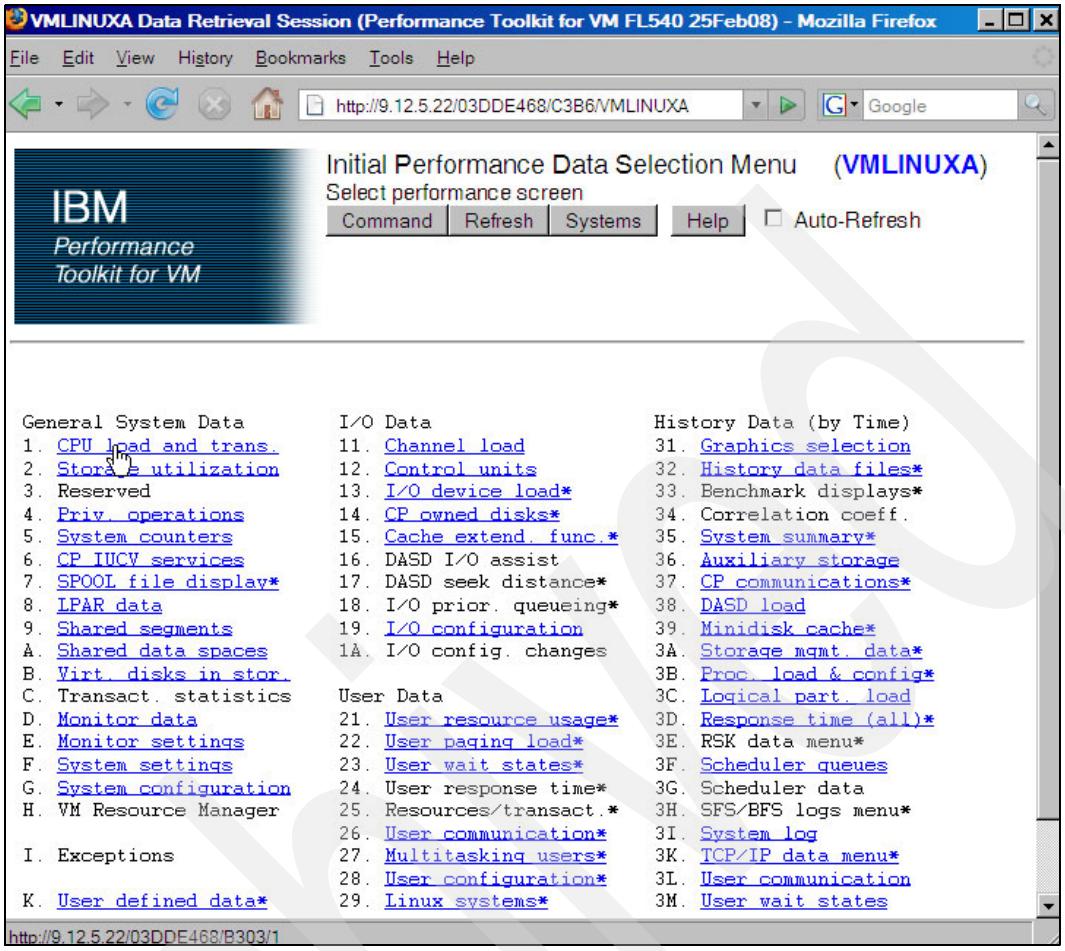


Figure 12-2 Browser interface to the Performance Toolkit

Using a 3270 interface

Logon to PERFSVM. Run the **PROFILE EXEC** and you should be put into the Performance Toolkit for z/VM environment. The subcommand **monitor** should present the following window.

```
==> profile
FCXBAS500I Performance Toolkit for VM FL530
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
...
FCX001                Performance Toolkit for VM
  FCXBAS500I Performance Toolkit for VM FL530
  HCPMOF6229E Monitor event collection is already active.
  HCPMOG6229E Monitor sample collection is already active.

Command ==> monitor
```

You will see the window displayed in Figure 12-3 on page 213.

```
FCX124          Performance Screen Selection  (FL540 25Feb08)    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. Reserved                13. I/O device load*       33. Benchmark displays*
4. Priv. operations        14. CP owned disks*        34. Correlation coeff.
5. System counters         15. Cache extend. func.*   35. System summary*
6. CP IUCV services        16. DASD I/O assist        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
```

Figure 12-3 Performance Toolkit 3270 Interface Main Menu window

Drilling down into report screens

You should now be able to use the active report screens. To drill down into these screens, move the cursor to any of the titles that are active (active titles display the number or letter in white; inactive titles are in green). Some of the more useful report screens to drill down into are:

```
21. User resource usage
22. User paging load
23. User wait states
28. User configuration
29. Linux systems
```

For example, to drill down into the Linux systems submenu window, enter the following command:

```
Command ===> 29
```

Then type S over the in the row corresponding to the report you want to see.

12.3 Monitoring Linux

Measurements can show resource consumption of the Linux guest as measured and dispatched by the VM host. It is also possible to 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 you want to monitor. There are different ways of gathering this data. Data can be gathered in the kernel and as a user application. Novell SUSE SLES9, SLES10

and Red Hat RHEL 4 and RHEL 5 have been enabled for the kernel to gather performance data. There is a package called the Linux RMF™ PM Data Gatherer (also called rmfpms) that runs as a user application. Both of these data gatherers work in conjunction with the IBM z/VM Performance Toolkit.

12.3.1 Monitoring Linux with rmfpms

As a user application, the Linux RMF PM Data Gatherer (rmfpms) can be used. Currently it is not part of an IBM product and is intended for evaluation purposes only. A description of rmfpms is as follows:

> rmfpms is a modular data gatherer for Linux. The gathered data can be analyzed using the RMF PM client application. The performance data is accessible through XML over HTTP so you can easily exploit it in your own applications.

The following Web site is a starting point:

http://www-03.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.html

To download the data gatherer, scroll down and look for the following text and links:

```
* 31 bit data gatherer (kernel24 - 630 KB, kernel26 - 1040 KB).
* 64 bit data gatherer (kernel24 - 650 KB, kernel26 - 666 KB).
```

You can download the appropriate gatherer through a browser, or if you have access to the Internet, you can use an FTP client. You will want one of two files, depending on whether you have a 31-bit or 64-bit kernel:

```
rmfpms_s390_kernel26.tgz - for 31-bit distributions
rmfpms_s390x_kernel26.tgz - for 64-bit distributions
```

If you cannot access Internet directly from your Linux image, download the tar file to your workstation and copy it to Linux. Following is an example of downloading the tar file for 64-bit distributions directly from the Internet. Get an SSH session on the controller or any other virtual server.

1. Change directory to /usr/local/src/:

```
# cd /usr/local/src
```

2. Download the appropriate tar file with the **wget** command. For a 64-bit SLES 9 and for SLES 10, use:

```
# wget ftp://ftp.software.ibm.com/eserver/zseries/zos/rmf/rmfpms_s390x_kernel26.tgz
--17:26:26--
ftp://ftp.software.ibm.com/eserver/zseries/zos/rmf/rmfpms_s390x_kernel26.tgz
           => `rmfpms_s390x_kernel26.tgz'
Resolving ftp.software.ibm.com... 9.17.252.40
Connecting to ftp.software.ibm.com[9.17.252.40]:21... connected.
Logging in as anonymous ... Logged in!
...
100%[====================================>] 1,097,125    538.93K/s
17:26:32 (537.12 KB/s) - `rmfpms_s390x_kernel26.tgz' saved [1097125]
```

3. Untar the file with the **tar** command and change to the rmfpms/ directory:

```
# tar xzf rmfpms_s390x_kernel26.tgz
# ls
rmfpms  rmfpms_s390x_kernel26.tgz
# cd rmfpms/
# ls
.rmfpms_config            README            bin  enable_autostart
.rmfpms_config_autostart  autostart_rmfpms  doc  rmfpms
```

4. You should now be able to start **rmfpms** in the bin/ directory with the following command:

```
# bin/rmfpms start
Creating /root/rmfpms/.rmfpms ...
Starting performance gatherer backends ...
 DDSRV: RMF-DDS-Server/Linux-Beta (Sep  8 2007) started.
 DDSRV: Functionality Level=2.339
 DDSRV: Reading exceptions from gpmexsys.ini and gpmexusr.ini.
DDSRV: Server will now run as a daemon process.
done!
```

5. After it is running, you can view the performance data from a browser pointing to the Linux image and port 8803, as shown in Figure 12-4.

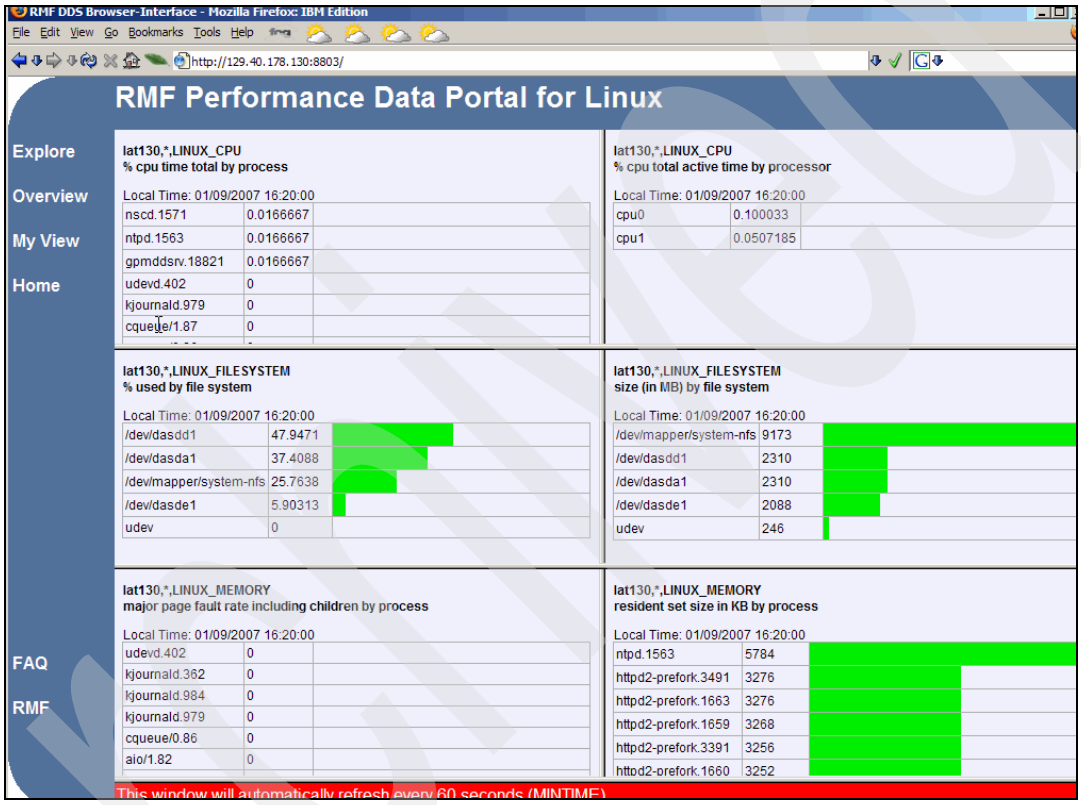


Figure 12-4 Browser view of rmfpms interface

You can also register Linux images with the Performance Toolkit, as described in 12.4, “Viewing Linux data in the Performance Toolkit” on page 216.

12.3.2 Monitoring Linux performance data from the kernel

To monitor Linux performance data directly from the kernel, the both of the following requirements 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 should be true because the OPTION APPLMON was set for the controller and for Linux user IDs in earlier sections.

For the second requirement, SLES 9, SLES 10 and RHEL 5 now have this function built in. Details of this function are described in Chapter 15, “Linux monitor stream support for z/VM”

in *Device Drivers, Features, and Commands* documentation for the October 2005 stream, which is available on the Web at:

http://www-128.ibm.com/developerworks/linux/linux390/october2005_documentation.html

A brief explanation of how to use this built-in monitoring function is given here.

1. There are three modules built into the kernel but not loaded, by default: `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
   ```

2. There is no output, which means that no modules with the string `appldata` are loaded. Load those modules now through the **`modprobe`** command and verify they have been loaded:

   ```
   # modprobe appldata_mem
   # modprobe appldata_os
   # modprobe appldata_net_sum
   ```

3. Now if you repeat the **`lsmod`** command, you should see the following:

   ```
   # lsmod | grep appldata
   appldata_net_sum       20064  0
   appldata_os            22536  0
   appldata_mem           20576  0
   ```

4. The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory there are five files:

`timer`	Controls whether any data gathering is in effect
`mem`	Controls the memory data gathering module
`os`	Controls the CPU data gathering module
`net_sum`	Controls the net data gathering module
`interval`	Sets the interval, in milliseconds, that samples will be taken

5. To turn on the built-in kernel monitoring, use the **`echo`** command to send a non-zero value into the **`timer`**, **`mem`**, **`os`** and **`net_sum`** 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

   # echo 20000 > /proc/sys/appldata/interval
   ```

   Built-in kernel monitoring should now be turned on.

Note: We do not recommend keeping Linux data gatherers running in all images all the time, because this may cause a serious performance problem. Instead, start gatherers only for specific periods of time and only when you have a performance problem to investigate. Also, as Linux monitoring data is captured, the Performance Toolkit's minidisk space can fill up relatively quickly.

12.4 Viewing Linux data in the Performance Toolkit

After the system has had some time to collect data, you should be able to use the Performance Toolkit to view Linux performance data. There should be entries in the `FCONX LINUXUSR` file as completed in 12.2.2, "Configuring Web Browser support" on page 207. To

view that data, drill down into menu 29, `Linux systems`. This can be done either from the browser interface or the 3270 interface as shown in Figure 12-5.

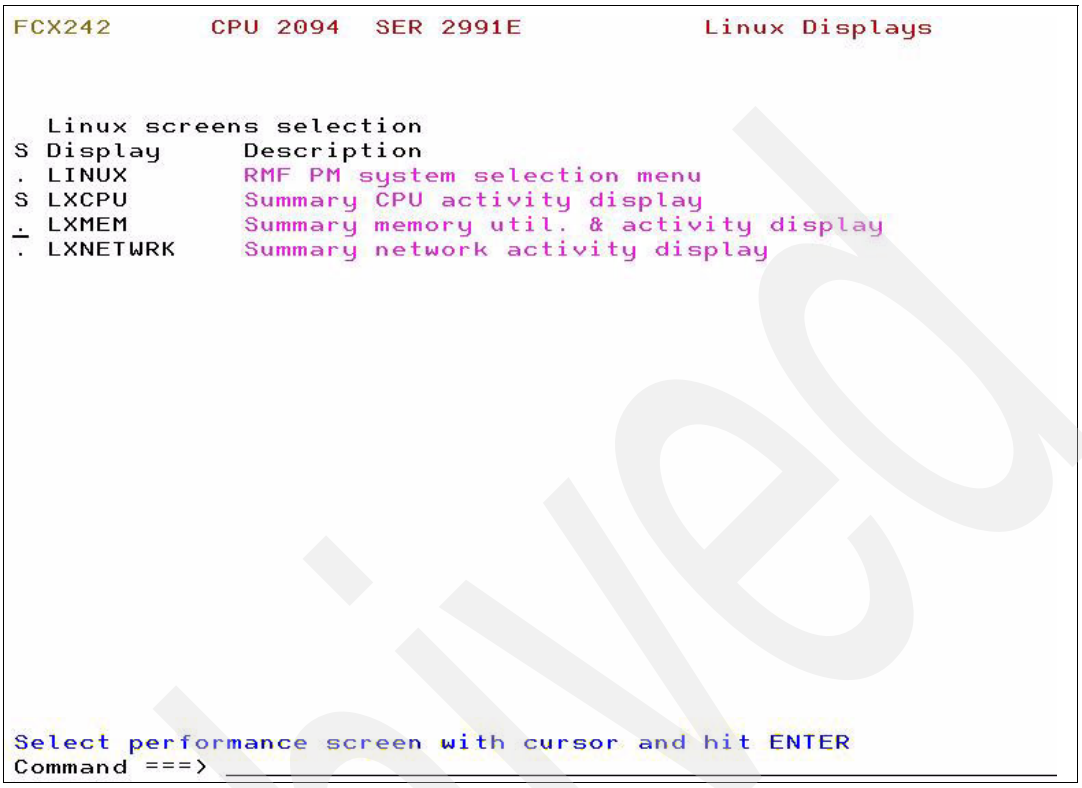


Figure 12-5 Linux Guest Systems submenu

Type `S` over the period on the left side of the submenu window in the row corresponding to the report you want to see. You should see a new report window with the Linux guest system CPU overview.

12.5 Common Performance Toolkit issue

When you browse performance toolkit screens, check the Performance Toolkit console log exceptions. If you get `FCXPMN446E Incomplete monitor data: DCSS size too small` or `FCXPMN446E Incomplete monitor data: SAMPLE CONFIG size too small` you have to act; otherwise, data in screens may be wrong or missing (that is, your data may be suspect).

To enlarge SAMPLE CONFIG or EVENT CONFIG, you may use or add the following commands to the PERFSVM PROFILE EXEC:

```
'CP MONITOR SAMPLE CONFIG SIZE xxx'
'CP MONITOR EVENT CONFIG SIZE yyy'
```

To enlarge DCSS, here is a hint. First, determine where the current MONDCSS is located by entering:

```
q nss name mondcss map
```

This results in output that is similar to:

```
FILE FILENAME FILETYPE MINSIZE BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
7271 MONDCSS CPDCSS N/A 03300 06FFF SC R 00007 N/A N/A
```

In this example, it starts at x3300 and ends at x6FFF (this is a PAGE boundary). You also need to issue **q nss all map** to determine where the new MONDCSS can be located with no overlap.

Important: Check carefully to make sure that the new MONDCSS does not overlap any other NSS. If it does, PERKIT may not be able to use it.

One way to test whether the MONDCSS segment can be loaded is to issue SEGMENT LOAD MONDCSS before you start PERFKIT. If it returns any error (for example, DMSDCS343E), there may be an overlap and you have to redefine the MONDCSS location.

Often, you can simply raise the ending location of the current MONDCSS to solve this problem.

First, delete the old MONDCSS by issuing:

```
PURGE NSS NAME MONDCSS
```

If any users are currently connected to the MONDCSS, you need to log off those users. You can determine who is connected by issuing **Q MONITOR**.

Next, create a new MONDCSS:

```
CP DEFSEG MONDCSS 03300-09FFF SC RSTD
```

Save the new MONDCSS:

```
CP SAVESEG MONDCSS
```

Recommendation: There is a relationship between the PERFSVM virtual machine storage size and the MONDCSS segment size, as described here.

- The PERFSVM virtual machine storage size should be of ample size to contain the entire MONDCSS segment, plus some CMS pointer tables.
- Or, the MONDCSS must be located outside the size of the PERFSVM virtual machine storage size (the preferred solution).

Then you can restart PERFKIT. If you receive any messages from the PERFKIT that the MONDCSS SAMPLE CONFIG size is too small, you have to enlarge it as previously explained. The same would apply to the MONITOR EVENT CONFIG SIZE, if you receive a PERFKIT message pertaining to that.

If you continue to receive MONDCSS SIZE TOO SMALL messages, you will need to restart this process over again, increasing the total size of the MONDCSS segment.

RESTART the PERFKIT and monitor the startup for at least 2 monitor interval timeframes to ensure things are working and there are no error type messages.

A

References

This appendix describes the location and content of z/VM configuration files, and provides quick reference sheets for the XEDIT and vi text editors.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

z/VM configuration files

z/VM differs from Linux in regard to the location and number of configuration files. In Linux, there are many configuration files and most of them are in or under the `/etc/` directory. On z/VM, there are relatively few configuration files. However, they are on many different minidisks.

Table 1 summarizes the location and content of z/VM configuration files.

Table 1 z/VM configuration files

File	Location	Description
SYSTEM CONFIG	MAINT CF1	This is the operating system's main configuration file. It defines the system name, the CP volumes, user volumes and other settings.
USER DIRECT	MAINT 2CC	This file defines the user directory. All user IDs or virtual machines known to the system are defined here (assuming a directory maintenance product is not being used).
<System_ID> TCPIP	TCPMAINT 198	This file defines the resources for the primary z/VM TCP/IP stack, including TCP/IP address, OSA resources, subnet mask and gateway. It is initially created by the IPWIZARD tool as `PROFILE TCPIP`.
SYSTEM DTCPARMS	TCPMAINT 198	This file is created to define the TCP/IP stacks on the system. It is initially created by the IPWIZARD tool.
TCPIP DATA	TCPMAINT 592	This file defines the DNS server, the domain name and some 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 up. It is analogous to the /etc/inittab file in Linux.

Quick reference sheets

This section contains quick reference sheets for the XEDIT and vi editors.

XEDIT quick reference sheet

XEDIT has line commands which are typed on the command line (===>) and prefix commands which are typed over the line numbers on the left side of the panel.

Line 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. '*' can be used for 'n' and 'm'
/<string>            Search for 'string' from the current line
-/<string>           Search backwards for 'string'
all /<string>/       Show all occurences 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 don't warn of overwrite
save                 Save the current file but don't exit
```

```
quit                  Exit XEDIT if no changes have been made
qquit                 Exit XEIDT even if changes have not been 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
:<n>                  Move to line ‘n’
?                     Display last command
=                     Execute last command
x <file>              Edit ‘file’ and put it into the XEDIT “ring”
x                     Move to the next file in the ring
```

Prefix commands

```
a      Add one line
a<n>   Add 'n' lines
c      Copies one line
cc     Copies a block of lines
d      Deletes one line
dd     Deletes 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
```

vi quick reference sheet

Following is a small, but commonly used, subset of vi commands.The vi editor has three modes:

1. Input mode - using the Insert key, **i**, **o** (add a line below), **0** (add a line above) and other commands puts you in this mode. When you are in this mode, you will see the text --INSERT-- in the last line.
2. Command mode - 'Esc' gets you out of input mode and into command mode.

```
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      do the last command again (this can be powerful)
jkl;   cursor movement
A      add text at the end of the line
<nn>G  go to line <nn>
G      go to the last line in the file
yy     yank a line (copy into buffer)
<n>yy  yank n lines
```

3. Command line mode - pressing the colon (**:**) key brings you to this mode.

```
:wq       save (write & quit)
:q!       quit and discard changes
```

Archived

```
:<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
```

Archived

B

z/VM source code

This appendix lists the source code associated with this book.

Obtaining and using the Web material

The pdf of this book is available on the Internet at the following address:

`http://www.redbooks.ibm.com/abstracts/sg247493.html`

The files associated with this book are in a gzipped tar file at the following address:

`ftp://www.redbooks.ibm.com/redbooks/SG247493`

Download the tar file to your NFS server and use it as is described in 7.5.1, “Copying files to the controller” on page 108. After untarring the file, you will have a directory named `virt-cookbook-S10SP2`.

Under that directory are the following files and directory:

`README.txt`	The main README file
`clone.sh`	The script to clone the golden image to a target user ID
`vm/`	A directory containing files used on z/VM

© Copyright IBM Corp. 2008. All rights reserved.

z/VM REXX EXECs and XEDIT macros

This section lists z/VM REXX EXECs and XEDIT macros.

The CPFORMAT EXEC

Following is the code for the EXEC that formats multiple disks through **CPFMTXA** (described in 4.6.1, “Formatting the paging volumes” on page 48):

```
/*+---------------------------------------------------------------------+*/
/*| EXEC: CPFORMAT - wrapper around CPFMTXA to format many DASD          |*/
/*|  retVal: 0 - success                                                |*/
/*|          1 - help was asked for or given                            |*/
/*|          2 - user is not sure                                       |*/
/*|          3 - DASD (minidisk) range is not valid                     |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT      |*/
/*+---------------------------------------------------------------------+*/
/* For details on how this EXEC is used, see one of the two  books:
"z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247493.html
 -or-
"z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5.2"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247492.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
---------------------------------------------------------------------*/
firstChar = 'M' /* change this for an LPAR ID other than 'M' */
parse upper arg dasds "AS " type
if ((dasds = '') | (dasds = '?')) then call help
labelPrefix = getLabelPrefix(firstChar type)
numDasd = parseDasd(dasds)
answer = areYouSure(type)
if (answer = 'Y') then  /* the user is sure */
do
  formatted = ""
  retVal = doFormat(labelPrefix numDasd type)
  call doReport retVal
end
else
  retVal = 2
exit retVal

/*+---------------------------------------------------------------------+*/
help: procedure
/*+---------------------------------------------------------------------+*/
  parse source . . fn .
  say ''
  say 'Synopsis:'
```

```
  say ''
  say '  Format one or a range of DASD as page, perm, spool or temp disk space'
  say '  The label written to each DASD is V<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 "                                              .-PAGE-."
  say "   >>--CPFORMAT--.-rdev--------------.--AS---+-PERM-+--------><"
  say "                 | <---------------< |       '-SPOL-'"
  say "                 '-rdev1-rdev2-------'  "
  say ''
exit 1

/*+---------------------------------------------------------------------+*/
areYouSure: procedure
/*| Show minidisks, ask are you sure                                    |*/
/*|  parm 1: type - PERM, PAGE, or SPOL                                 |*/
/*|  retVal: firstChar - LPAR identifier, 'V' by default                |*/
/*+---------------------------------------------------------------------+*/
  arg type
  say ''
  say 'WARNING - this will destroy data!'
  say 'ARE YOU SURE you want to format the DASD as' type 'space (y/n)?'
  parse upper pull answer
return  substr(answer, 1, 1) /* from areYouSure */

/*+---------------------------------------------------------------------+*/
getLabelPrefix: procedure
/*| Return first two chararcters of label                               |*/
/*|  parm 1: firstChar - LPAR identifier, 'V' by default                |*/
/*|  retVal: the two character label prefix                             |*/
/*+---------------------------------------------------------------------+*/
  arg firstChar type
  select
  when (type = PERM) then
    labelPrefix = firstChar||'M' /* for VM Minidisk */
  when (type = PAGE) then
    labelPrefix = firstChar||'P' /* for VM Page */
  when (type = SPOL) then
    labelPrefix = firstChar||'S' /* for VM Spool */
  otherwise
    do
      say 'Error: "AS" must be present, type must be PERM, PAGE or SPOL'
      call help
    end /* otherwise */
  end /* select */
return labelPrefix /* from getLabelPrefix */

/*+---------------------------------------------------------------------+*/
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
  say ''
  say 'Format the following DASD:'
  do while (dasds <> '')
```

Archived

```
    parse upper var dasds dasd dasds
    dashPos = pos('-', dasd)
    if (dashPos = 0) then  /* there is just one DASD */
    do
      numDasd = numDasd + 1
      dasdList.numDasd = dasd
      'CP Q MDISK' dasdList.numDasd 'LOCATION'
      if (rc <> 0) then
      do
        say 'Return code from Q MDISK =' rc
        say 'Are all DASD ATTached?'
        exit 3
      end
      call checkReserved(dasdList.numDasd)
    end /* do */
    else /* process the range of DASD */
    do
      startRange = substr(dasd, 1, dashPos - 1)
      endRange = substr(dasd, dashPos + 1, length(dasd) - dashPos)
      do i = x2d(startRange) to x2d(endRange)
        numDasd = numDasd + 1
        dasdList.numDasd = d2x(i)
        'CP Q MDISK' dasdList.numDasd 'LOCATION'
        if (rc <> 0) then
        do
          say 'Return code from Q MDISK =' rc
          exit 3
        end
        call checkReserved(dasdList.numDasd)
      end /* do i */
    end /* else */
  end /* do while */
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
  'CP TERM MORE 1 1'
  do i = 1 to numDasd
    label = getLabel(labelPrefix dasdList.i)
    retVal = formatOne(dasdList.i type label)
    if (retVal ^= 0) then
    do
      say "Error from CPFMTXA on DASD" label "rc =" retVal
      leave /* error - abort! */
    end
    formatted = formatted label
  end /* do i = */
  'CP TERM MORE 50 10'
return retVal /* from doFormat */

/*+---------------------------------------------------------------------+*/
checkReserved: procedure
/*| Try copying an already formatted DASD then relabelling it           |*/
```

```
/*|  parm 1: source                                                    |*/
/*|  parm 2: target                                                    |*/
/*|  parm 3: label                                                     |*/
/*+--------------------------------------------------------------------+*/
  arg dasd
  /* create a list of reserved dasd - this is somewhat hokey to be sure
     but it's better to be hokey than to format system minidisks! */
  resvd1 = "0122 0123 0124 0125 019O 0191 0193 0194 019D 019E 02O1 02A2"
  resvd2 = "02A4 02A6 02C2 02C4 02CC 02D2 0319 03A2 03A4 03A6 03B2 03C2"
  resvd3 = "03C4 03D2 04OO 04O1 04O2 0405 049O 0493 049B 049E 04A2 04A4"
  resvd4 = "04A6 04B2 04C2 04C4 04D2 05OO 051D 05A2 05A4 05A6 05B2 05C2"
  resvd5 = "05C4 05D2 05E5 05E6 06A2 06A4 06A6 06B2 06C2 06C4 06D2 07A2"
  resvd6 = "07A4 07A6 07B2 07C2 07C4 07D2 0CF1 0CF2 0CF3"
  reserved = resvd1 resvd2 resvd3 resvd4 resvd5 resvd6
  if (index(reserved, dasd) <> 0) then /* MAINT minidisk - ABORT! */
  do
    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
  'DETACH' dasds
  'ATTACH' dasds '*'
  say ''
  say 'DASD status after:'
  'CP Q 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 - PAGE, SPOL or PERM                                 |*/
/*|  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'
  'CPFMTXA'
  retVal = rc
```

Archived

```
return retVal /* from formatOne */

/*+---------------------------------------------------------------+*/
getLabel: procedure
/*| Compose the six character label of a minidisk                 |*/
/*|  parm 1: labelPrefix - first two characters of label          |*/
/*|  parm 2: disk - vaddr of length 1, 2, 3 or 4                  |*/
/*|  return: the 6 character label                                |*/
/*+---------------------------------------------------------------+*/
  arg labelPrefix disk
  diskLen = length(disk)
  select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
  end /* select */
return label /* from getLabel */
```

The CHPW540 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.4 USER DIRECT file:

```
/*+---------------------------------------------------------------+*/
/* CHPW540 XEDIT - change all passwords in z/VM 5.2 USER DIRECT file */
/*+---------------------------------------------------------------+*/
/* For details on how this macro is used, see the book:
"z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247493.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
---------------------------------------------------------------*/

parse arg fn ft fm '(' options ')' newPass .
if (length(newPass) > 8) then
do
  say "Error: new password must be 8 characters or fewer"
  exit
end
say ''
say 'Changing all passwords to:' newPass
say ''

/* set some values */
'command set stay on'
```

```
'command set num on'
'command set nulls on'
'command set serial off'
'command set cmdline bottom'
'command set curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'

/* change user ID passwords */
'command c/MAINT MAINT/MAINT' newPass'/*'
'command c/AVSVM AVSVM/AVSVM' newPass'/*'
'command c/TSAFVM TSAFVM/TSAFVM' newPass'/*'
'command c/GCS GCS/GCS' newPass'/*'
'command c/GCSXA GCSXA/GCSXA' newPass'/*'
'command c/AUDITOR AUDITOR/AUDITOR' newPass'/*'
'command c/AUTOLOG1 AUTOLOG1/AUTOLOG1' newPass'/*'
'command c/AUTOLOG2 AUTOLOG2/AUTOLOG2' newPass'/*'
'command c/BLDCMS  BLDCMS/BLDCMS' newPass'/*'
'command c/BLDNUC  BLDNUC/BLDNUC' newPass'/*'
'command c/BLDRACF  BLDRACF/BLDRACF' newPass'/*'
'command c/BLDSEG BLDSEG/BLDSEG' newPass'/*'
'command c/CMS1  CMS1/CMS1' newPass'/*'
'command c/CMSBATCH CMSBATCH/CMSBATCH' newPass'/*'
'command c/DISKACNT DISKACNT/DISKACNT' newPass'/*'
'command c/EREP EREP/EREP' newPass'/*'
'command c/IBMUSER IBMUSER/IBMUSER' newPass'/*'
'command c/LGLOPR  LGLOPR/LGLOPR' newPass'/*'
'command c/MIGMAINT MIGMAINT/MIGMAINT' newPass'/*'
'command c/MONWRITE MONWRITE/MONWRITE' newPass'/*'
'command c/OP1 OP1/OP1' newPass'/*'
'command c/OPERATNS OPERATNS/OPERATNS' newPass'/*'
'command c/OPERATOR OPERATOR/OPERATOR' newPass'/*'
'command c/OPERSYMP OPERSYMP/OPERSYMP' newPass'/*'
'command c/SYSADMIN SYSADMIN/SYSADMIN' newPass'/*'
'command c/SYSDUMP1 SYSDUMP1/SYSDUMP1' newPass'/*'
'command c/SYSMAINT SYSMAINT/SYSMAINT' newPass'/*'
'command c/SYSMON  SYSMON/SYSMON' newPass'/*'
'command c/VMRMADMN VMRMADMN/VMRMADMN' newPass'/*'
'command c/VMRMSVM VMRMSVM/VMRMSVM' newPass'/*'
'command c/VMSERVR VMSERVR/VMSERVR' newPass'/*'
'command c/VMSERVS VMSERVS/VMSERVS' newPass'/*'
'command c/VMSERVU VMSERVU/VMSERVU' newPass'/*'
'command c/VMUTIL VMUTIL/VMUTIL' newPass'/*'
'command c/VSMPROXY VSMPROXY/VSMPROXY' newPass'/*'
'command c/VSMREQIN VSMREQIN/VSMREQIN' newPass'/*'
'command c/VSMREQIU VSMREQIU/VSMREQIU' newPass'/*'
'command c/VSMSERVE VSMSERVE/VSMSERVE' newPass'/*'
'command c/VSMWORK1 VSMWORK1/VSMWORK1' newPass'/*'
'command c/VSMWORK2 VSMWORK2/VSMWORK2' newPass'/*'
'command c/VSMWORK3 VSMWORK3/VSMWORK3' newPass'/*'
'command c/ZVMMAPLX MAINT/ZVMMAPLX' newPass'/*'
'command c/5684042J 5684042J/5684042J' newPass'/*'
'command c/4OSASF4O 4OSASF4O/4OSASF4O' newPass'/*'
'command c/OSADMIN1 OSADMIN1/OSADMIN1' newPass'/*'
'command c/OSADMIN2 OSADMIN2/OSADMIN2' newPass'/*'
'command c/OSADMIN3 OSADMIN3/OSADMIN3' newPass'/*'
'command c/OSAMAINT OSAMAINT/OSAMAINT' newPass'/*'
```

```
'command c/OSASF OSASF/OSASF' newPass'/*'
'command c/5VMRSC4O 5VMRSC4O/5VMRSC4O' newPass'/*'
'command c/RSCS RSCS/RSCS' newPass'/*'
'command c/RSCSAUTH RSCSAUTH/RSCSAUTH' newPass'/*'
'command c/RSCSDNS RSCSDNS/RSCSDNS' newPass'/*'
'command c/XCHANGE XCHANGE/XCHANGE' newPass'/*'
'command c/5VMTCP4O 5VMTCP4O/5VMTCP4O' newPass'/*'
'command c/TCPIP TCPIP/TCPIP' newPass'/*'
'command c/TCPMAINT TCPMAINT/TCPMAINT' newPass'/*'
'command c/ADMSERV ADMSERV/ADMSERV' newPass'/*'
'command c/DHCPD DHCPD/DHCPD' newPass'/*'
'command c/DTCVSW1  DTCVSW1/DTCVSW1' newPass'/*'
'command c/DTCVSW2  DTCVSW2/DTCVSW2' newPass'/*'
'command c/FTPSERVE FTPSERVE/FTPSERVE' newPass'/*'
'command c/IMAP IMAP/IMAP' newPass'/*'
'command c/IMAPAUTH IMAPAUTH/IMAPAUTH' newPass'/*'
'command c/LDAPSRV  LDAPSRV/LDAPSRV' newPass'/*'
'command c/LPSERVE LPSERVE/LPSERVE' newPass'/*'
'command c/MPROUTE MPROUTE/MPROUTE' newPass'/*'
'command c/NAMESRV NAMESRV/NAMESRV' newPass'/*'
'command c/NDBPMGR NDBPMGR/NDBPMGR' newPass'/*'
'command c/NDBSRVO1 NDBSRVO1/NDBSRVO1' newPass'/*'
'command c/PORTMAP PORTMAP/PORTMAP' newPass'/*'
'command c/REXECD REXECD/REXECD' newPass'/*'
'command c/RXAGENT1 RXAGENT1/RXAGENT1' newPass'/*'
'command c/SMTP SMTP/SMTP' newPass'/*'
'command c/SNALNKA SNALNKA/SNALNKA' newPass'/*'
'command c/SNMPD SNMPD/SNMPD' newPass'/*'
'command c/SNMPQE SNMPQE/SNMPQE' newPass'/*'
'command c/SNMPSUBA SNMPSUBA/SNMPSUBA' newPass'/*'
'command c/SSLSERV SSLSERV/SSLSERV' newPass'/*'
'command c/TFTPD TFTPD/TFTPD' newPass'/*'
'command c/UFTD    UFTD/UFTD' newPass'/*'
'command c/VMKERB VMKERB/VMKERB' newPass'/*'
'command c/VMNFS VMNFS/VMNFS' newPass'/*'
'command c/X25IPI X25IPI/X25IPI' newPass'/*'
'command c/5VMDIR4O 5VMDIR4O/5VMDIR4O' newPass'/*'
'command c/5VMRAC4O 5VMRAC4O/5VMRAC4O' newPass'/*'
'command c/RACFSMF RACFSMF/RACFSMF' newPass'/*'
'command c/RACFVM RACFVM/RACFVM' newPass'/*'
'command c/RACMAINT RACMAINT/RACMAINT' newPass'/*'
'command c/5VMPTK4O 5VMPTK4O/5VMPTK4O' newPass'/*'
'command c/PERFSVM  PERFSVM/PERFSVM' newPass'/*'
'command c/5VMHCD4O 5VMHCD4O/5VMHCD4O' newPass'/*'
'command c/CBDIODSP CBDIODSP/CBDIODSP' newPass'/*'
'command c/LNXMAINT LNXMAINT/LNXMAINT' newPass'/*'

/* change mindisk passwords */
'command c/ALL      WRITE    MULTIPLE/ALL' newPass newPass'/*'
'command c/ALL      WTCPMAIN MTCPMAIN/ALL' newPass newPass'/*'
'command c/RADMSERV WADMSERV MADMSERV/'newPass newPass newPass'/*'
'command c/RAUDITOR WAUDITOR MAUDITOR/'newPass newPass newPass'/*'
'command c/RAUTOLOG WAUTOLOG MAUTOLOG/'newPass newPass newPass'/*'
'command c/RAVSOBJ  WAVSOBJ  MAVSOBJ/'newPass newPass newPass'/*'
'command c/RBATCH   WBATCH   MBATCH/'newPass newPass newPass'/*'
'command c/RCATALOG WCATALOG/'newPass newPass'/*'
'command c/RCONTROL WCONTROL/'newPass newPass'/*'
'command c/RCRRLOG1 WCRRLOG1/'newPass newPass'/*'
'command c/RDATA    WDATA/'newPass newPass'/*'
'command c/RDHCPD   WDHCPD   MDHCPD/'newPass newPass newPass'/*'
```

```
'command c/RDVF     WDVF     MDVF/'newPass newPass newPass'/*'
'command c/READ     WRITE    MULTIPLE/'newPass newPass newPass'/*'
'command c/READ     WRITE/'newPass newPass'/*'
'command c/RFTPSERV WFTPSERV MFTPSERV/'newPass newPass newPass'/*'
'command c/RGCS     WGCS     MGCS/'newPass newPass newPass'/*'
'command c/RIMAP    WIMAP    MIMAP/'newPass newPass newPass'/*'
'command c/RLOG1    WLOG1/'newPass newPass newPass'/*'
'command c/RLOG2    WLOG2/'newPass newPass newPass'/*'
'command c/RLPSERVE WLPSERVE MLPSERVE/'newPass newPass newPass'/*'
'command c/RMAINT   WMAINT   MMAINT/'newPass newPass newPass'/*'
'command c/RMPROUTE WMPROUTE MMPROUTE/'newPass newPass newPass'/*'
'command c/RNAMESRV WNAMESRV MNAMESRV/'newPass newPass newPass'/*'
'command c/RNDBPMGR WNDBPMGR MNDBPMGR/'newPass newPass newPass'/*'
'command c/RNDBSRVO WNDBSRVO MNDBSRVO/'newPass newPass newPass'/*'
'command c/RPORTMAP WPORTMAP MPORTMAP/'newPass newPass newPass'/*'
'command c/RREXECD  WREXECD  MREXECD/'newPass newPass newPass'/*'
'command c/RSERVER  WSERVER/'newPass newPass'/*'
'command c/RSMTP    WSMTP    MSMTP/'newPass newPass newPass'/*'
'command c/RSNALNKA WSNALNKA MSNALNKA/'newPass newPass newPass'/*'
'command c/RSNMPD   WSNMPD   MSNMPD/'newPass newPass newPass'/*'
'command c/RSNMPQE  WSNMPQE  MSNMPQE/'newPass newPass newPass'/*'
'command c/RSSLSERV WSSLSERV MSSLSERV/'newPass newPass newPass'/*'
'command c/RSYSMON  WSYSMON  MSYSMON/'newPass newPass newPass'/*'
'command c/RTCPIP   WTCPIP   MTCPIP/'newPass newPass newPass'/*'
'command c/RTCPMAIN WTCPMAIN MTCPMAIN/'newPass newPass newPass'/*'
'command c/RTFTPD   WTFTPD   MTFTPD/'newPass newPass newPass'/*'
'command c/RTSAFOBJ WTSAFOBJ MTSAFOBJ/'newPass newPass newPass'/*'
'command c/RUFTD    WUFTD    MUFTD/'newPass newPass newPass'/*'
'command c/RVMKERB  WVMKERB  MVMKERB/'newPass newPass newPass'/*'
'command c/RVMNFS   WVMNFS   MVMNFS/'newPass newPass newPass'/*'
'command c/RX25IPI  WX25IPI  MX25IPI/'newPass newPass newPass'/*'
'command c/R4TCPIP  W4TCPIP  M4TCPIP/'newPass newPass newPass'/*'
'command c/RDTCVSW1 WDTCVSW1 MDTCVSW1/'newPass newPass newPass'/*'
'command c/RDTCVSW2 WDTCVSW2 MDTCVSW2/'newPass newPass newPass'/*'
'command c/RCRRLOG2 WCRRLOG2/'newPass newPass'/*'
'command c/RLDAPSRV WLDAPSRV MLDAPSRV/'newPass newPass newPass'/*'
'command c/RSNMPSUB WSNMPSUB MSNMPSUB/'newPass newPass newPass'/*'
'command c/MR READ/'MR newPass'/*'
```

The LABEL540 EXEC

Following is the code for the EXEC that changes the system labels of a z/VM 5.4 system:

```
/*+---------------------------------------------------------------+*/
/*| EXEC: LABEL540  wrapper around CPFMTXA to LABEL and ALLOC DASD |*/
/*|  retVal: 0 - success                                          |*/
/*|          1 - help was asked for or given                      |*/
/*|          2 - user is not sure                                 |*/
/*|          3 - DASD (minidisk) range is not valid               |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT|*/
/*+---------------------------------------------------------------+*/
/* For details on how this EXEC is used, see the book:
"z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247493.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
---------------------------------------------------------------------*/

/* Construct the two character label prefix */
firstChar = 'M' /* change this for an LPAR ID other than 'M' */
labelPrefix = firstChar'V'

/* Parse arguments */
parse upper arg res spl pag w01 w02 .
if (w02 = '') then call help


/* Construct the 5 labels */
resLabel = getLabel(labelPrefix res)
splLabel = getLabel(labelPrefix spl)
pagLabel = getLabel(labelPrefix pag)
w01Label = getLabel(labelPrefix w01)
w02Label = getLabel(labelPrefix w02)

/* Ask "Are you sure?" */
say 'The volumes are:'
'CP Q' res spl pag w01 w02
say ''
say 'The system volume labels will become:'
say resLabel splLabel pagLabel w01Label w02Label
say ''
say 'ARE YOU SURE you want to relabel the DASD (y/n)?'
parse upper pull answer
ansFirstChar = substr(answer, 1, 1)
if (ansFirstChar ^= 'Y') then exit 2

/* Label the 4 volumes: RES is 123, W01 is 124, W02 is 125, SPL is 122 */
'CP TERM MORE 1 1'
'CPFMTXA 123' resLabel 'LABEL'
'CPFMTXA 124' w01Label 'LABEL'
'CPFMTXA 125' w02Label 'LABEL'
'CPFMTXA 122' splLabel 'LABEL'

/* LINK the 540PAG volume which is $PAGE$ A03, label it, DETACH it */
'CP LINK $PAGE$ A03 A03 MR'
'CPFMTXA A03' pagLabel 'LABEL'
'CP DET A03'
'CP TERM MORE 50 10'
exit

/*+---------------------------------------------------------------------+*/
help: procedure expose firstChar
/*+---------------------------------------------------------------------+*/
  parse source . . fn .
  say ""
  say "Synopsis:"
  say ""
  say "Relabel 5 system volumes (540RES, 540W01, ...) to" firstChar"V<xxxx>"
  say "   where <xxxx> is the 4 digit address"
  say ""
```

```
  say "Syntax is:"
  say ""
  say "   >>---LABEL540--res--spl--pag--w01--w02----------------------><"
  say ""
  say " where res, spl, pag, w01 and w02 are 4 digit virtual addresses"
  say " of the volumes that z/VM 5.2 is installed onto"
  say ""
exit 1

/*+---------------------------------------------------------------------+*/
getLabel: procedure
/*| Compose the six character label of a minidisk                       |*/
/*|  parm 1: labelPrefix - first two characters of label                |*/
/*|  parm 2: disk - vaddr of length 1, 2, 3 or 4                        |*/
/*|  return: the 6 character label                                      |*/
/*+---------------------------------------------------------------------+*/
  arg labelPrefix disk
  if (DATATYPE(disk, 'X') = 0) then
  do
    say "Error:" disk "is not a hexadecimal number"
    call help
  end
  diskLen = length(disk)
  select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
  end /* select */
return label /* from getLabel */
```

The LABEL540 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.4 USER DIRECT file:

```
/*+---------------------------------------------------------------------+*/
/*| EXEC: LABEL540  wrapper around CPFMTXA to LABEL and ALLOC DASD       |*/
/*|  retVal: 0 - success                                                |*/
/*|          1 - help was asked for or given                            |*/
/*|          2 - user is not sure                                       |*/
/*|          3 - DASD (minidisk) range is not valid                     |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT      |*/
/*+---------------------------------------------------------------------+*/
/* For details on how this EXEC is used, see the book:
"z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247493.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,
```

Archived

```
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
---------------------------------------------------------------------*/

firstChar = 'M' /* change this for an LPAR ID other than 'M' */
parse upper arg res spl pag w01 w02 .

parse upper arg fn ft fm '(' options ')' res spl pag w01 w02 .
if (w02 = '') then
do
  say "5 arguments required - exiting"
  exit
end
'command set stay on'
'command set num on'
'command set nulls on'
'command set serial off'
'command set cmdline bottom'
'command set curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'
'command top'
'command c/540RES/'firstChar'V'res'/*'
'command c/540W01/'firstChar'V'w01'/*'
'command c/540W02/'firstChar'V'w02'/*'
'command c/540SPL/'firstChar'V'spl'/*'
'command c/540PAG/'firstChar'V'pag'/*'
```

PROFILE EXEC for Linux user IDs

This section lists the code for the PROFILE EXEC that is shared among Linux user IDs 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 /* else */
```

C

Linux source code

This appendix provides the Linux source code (**clone.sh** script) associated with this book.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

The clone.sh script

Following is the **clone.sh** script:

```
#!/bin/bash
#
# 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 Virtualization Cookbook for SLES 10 SP2"
#  on the Web at: http://www.redbooks.ibm.com/abstracts/SG247493.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
# ----------------------------------------------------------------------------

#+-----------------------------------------------------------------------------+
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 sles10s2 to linux01"
  exit 1
 }

#+-----------------------------------------------------------------------------+
function processArguments()
# Parse command line arguments
# Args: The arguments passed in to the script
#+-----------------------------------------------------------------------------+
 {
  verbose="off"
  sourceID="none"
  targetID="none"
  while (( "$#" )); do
    case $1 in
      -v|--verbose)
         verbose="on"
         ;;
      from)
        shift
        sourceID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold source ID to upper case
        ;;
      to)
```

```
        shift
        targetID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold target ID to upper case
        ;;
    esac
    shift
  done
  if [ $sourceID = "none" ]; then # source user ID was not passed
    echo "Error: Source Linux user ID not supplied"
    help
  fi
  if [ $targetID = "none" ]; then # target user ID was not passed
    echo "Error: Target Linux user ID not supplied"
    help
  fi
 }

#+-------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+-------------------------------------------------------------------------+
 {
  echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
  return $retVal
 }

#+-------------------------------------------------------------------------+
function checkID()
# Verify user ID exists and is logged off
#    Arg 1: The user ID to check
#+-------------------------------------------------------------------------+
 {
  userID=$1
  echo "Checking that $userID exists and is not logged on ..."
  CPcmd QUERY $userID
  rc=$?
  case $rc in
    0)  # user ID is logged on or disconnected
      echo "$userID user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$userID user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "Return code of $rc unexpected from QUERY $userID"
      echo "User ID must exist and be logged off"
      exit 4
  esac
 }
```

```
#+-------------------------------------------------------------------------+
function prepareIPaddr()
# Set the variable "newIPaddr" by adding a backslash before any "."s
#    Arg 1: The IP address to be modified
#+-------------------------------------------------------------------------+
 {
  newIPaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }

#+-------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+-------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+-------------------------------------------------------------------------+
function copyDisk()
# Use FLASHCOPY to copy a disk, if it fails, fall back to dasdfmt then dd
#   Arg 1: Source vaddr
#   Arg 2: Target vaddr
#+-------------------------------------------------------------------------+
 {
  source=$1
  target=$2
  echo ""
  echo "FLASHCOPYing $source to $target ..."
  CPcmd FLASHCOPY $source 0 end to $target 0 end
  if [ $? != 0 ]; then
    echo "FLASHCOPY failed, falling back to dasdfmt and dd ..."
    chccwdev -e $source
    if [ $? != 0 ]; then exit 7; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 8; fi
    sleep 1
    srcDev=/dev/$(egrep ^0.0.$source /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 5; fi
    tgtDev=/dev/$(egrep ^0.0.$target /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 6; fi
    echo "dasdfmt-ing $tgtDev ..."
    dasdfmt -y -b 4096 -f $tgtDev
    if [ "$?" != 0 ]; then exit 9; fi
    echo "dd-ing $srcDev to $tgtDev ..."
    dd bs=4096 if=$srcDev of=$tgtDev
    if [ "$?" != 0 ]; then exit 10; fi
    echo "disabling and re-enabling $target ..."
    chccwdev -d $target
    if [ $? != 0 ]; then exit 11; fi
    chccwdev -e $target
```

Archived

```
    if [ $? != 0 ]; then exit 12; fi
  fi
 }

#+-------------------------------------------------------------------------+
function getNetworkInfo()
# Bring 191 minidisk online to check for two 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
  sleep 1 # TODO: get the command that waits until the disk is enabled.
  is=`lsdasd | grep 0191 | awk '{ print $6 }'`
  if [ "$is" = "is" ]; then # dasdN field is number 7
    CMSdisk=`lsdasd | grep 0191 | awk '{ print $7 }'`
  else # assume it is 6th field because space was eaten up by minor > 100
    CMSdisk=`lsdasd | grep 0191 | awk '{ print $6 }'`
  fi
  cmsfslst -d /dev/$CMSdisk | grep -i $sourceID | grep PARM-S10
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $sourceID PARM-S10 not found on 191 minidisk. Exiting"
    exit 14
  fi
  cmsfslst -d /dev/$CMSdisk | grep -i $targetID | grep PARM-S10
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $targetID PARM-S10 not found on 191 minidisk. Exiting"
    exit 15
  fi

# get informaton about target
  export local $(cmsfscat -a -d /dev/$CMSdisk $targetID.PARM-S10)
  tagetHostname=$Hostname
  targetIP=$HostIP
  targetDNS=$Nameserver
  targetGW=$Gateway
  targetMask=$Netmask
  targetBroadcast=$Broadcast
  prepareVaddr $ReadChannel
  targetReaddev=$newVaddr
  prepareVaddr $WriteChannel
  targetWritedev=$newVaddr
  prepareVaddr $DataChannel
  targetDatadev=$newVaddr

# get information about source
  export local $(cmsfscat -a -d /dev/$CMSdisk $sourceID.PARM-S10)
  sourceHostname=$Hostname
  prepareIPaddr $HostIP
  sourceIP=$newIPaddr
  prepareIPaddr $Nameserver
  sourceDNS=$newIPaddr
  prepareIPaddr $Gateway
```

Archived

```
  sourceGW=$newIPaddr
  prepareIPaddr $Netmask
  sourceMask=$newIPaddr
  prepareIPaddr $Broadcast
  sourceBroadcast=$newIPaddr
  prepareVaddr $ReadChannel
  sourceReaddev=$newVaddr
  prepareVaddr $WriteChannel
  sourceWritedev=$newVaddr
  prepareVaddr $DataChannel
  sourceDatadev=$newVaddr
 }

#+-------------------------------------------------------------------------+
function askAreYouSure()
# Ask "Are you sure?" - if not, then exit
#+-------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: Mindisks 100 and 101 will be copied to $targetID"
  echo "New host name will be:  $tagetHostname"
  echo "New IP address will be: $targetIP"
  echo "Other network data is retrieved from $targetID PARM-S10 on 191 disk"
  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 mindisks 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
  CPcmd det 1100
  # don't detach 2100 yet because it has to be modified

  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
```

```
  CPcmd det 1101
  CPcmd det 2101
 }

#+-------------------------------------------------------------------------+
function modifyClone()
# Mount newly copied system and modify networking info
#+-------------------------------------------------------------------------+
 {
  targetVaddr=2100
  targetDir="/mnt/targetLinux"         # directory of target Linux system
  echo "Mounting newly cloned image over $targetDir ..."
  if [ ! -d $targetDir ]; then
    mkdir $targetDir
    rc=$?
    if [ $rc != 0 ]; then
      echo "Error: mkdir $targetDir failed with $rc"
      exit 21
    fi
  fi
  # recycle target mount point
#  chccwdev -d $targetVaddr
  chccwdev -e $targetVaddr
  rc=$?
  if [ $rc != 0 ]; then
    echo "Fatal error: chccwdev -e $targetVaddr failed with $rc"
    CPcmd DET $targetVaddr
    exit 22
  fi
  sleep 1
  tDev=/dev/$(egrep ^0.0.$targetVaddr /proc/dasd/devices | awk '{ print $7 }')1
    if [ "$?" != 0 ]; then exit 23; fi
  echo "Mounting $tDev over $targetDir ..."
  mount $tDev $targetDir
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: 'mount $tDev $targetDir' failed with $rc"
    lsdasd
    CPcmd DET $targetVaddr
    exit 24
  fi
  echo "Modifying cloned image under $targetDir ..."
  sed --in-place -e "s/$sourceHostname/$tagetHostname/g" \
    $targetDir/etc/HOSTNAME
  sed --in-place -e "s/$sourceIP/$targetIP/g" \
    -e "s/${sourceHostname%%.*}/${tagetHostname%%.*}/g" $targetDir/etc/hosts
  sed --in-place -e "s/$sourceGW/$targetGW/g" \
    $targetDir/etc/sysconfig/network/routes
  sed --in-place -e "s/$sourceIP/$targetIP/g" \
    -e "s/$sourceMask/$targetMask/g" \
    -e "s/$sourceBroadcast/$targetBroadcast/g" \
    $targetDir/etc/sysconfig/network/ifcfg-qeth-bus-ccw-$targetReaddev
  sed --in-place -e "s/$sourceDNS/$targetDNS/g" $targetDir/etc/resolv.conf

# Regenerate SSH keys
  echo "Regenerating SSH keys in $targetDir/etc/ssh/ ..."
  rm $targetDir/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $targetDir/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $targetDir/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $targetDir/etc/ssh/ssh_host_key
```

Archived

```
# Remove any old entry, then copy clone's public key to known_hosts file
  echo "Adding $targetIP to known_hosts file"
  cd /root/.ssh
  grep -v $targetIP known_hosts > known_hosts.temp
  mv known_hosts.temp known_hosts

# construct known_hosts entry with three fields: IP@ "ssh-rsa" clones_public_key
  newKey=`cat $targetDir/etc/ssh/ssh_host_rsa_key.pub | awk '{ print $2 }'`
  echo "$targetIP ssh-rsa $newKey" >> known_hosts

# clean up
  sleep 1
  sync # sync disks
  umount $targetDir
  chccwdev -d $targetVaddr
  CPcmd DETACH $targetVaddr
  return 0
 }

# 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
CPcmd XAUTOLOG $targetID                  # bring new clone to life
if [ $verbose = "on" ]; then set +vx; fi  # turn off debug
echo "Successfully cloned $sourceID to $targetID"
echo "You should be able to ping $targetIP within one minute"
exit 0
```

Archived

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 publications

For information about ordering these publications, see “How to get Redbooks” on page 244. Note that some of the documents referenced here may be available in softcopy only

- *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
- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864
- *Running Linux Guest in less than CP Privilege Class G*, REDP-3870

Other publications

- *Linux for zSeries and S/390 Device Drivers, Features, and Commands*, LNUX-1403

  `http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/l26cdd04.pdf`
- *SUSE LINUX Enterprise Server* documentation - on CD1 or DVD1 in `docu/` directory:
  - *SUSE LINUX Enterprise Server* INSTALLATION AND ADMINISTRATION
  - SUSE LINUX Enterprise Server: ARCHITECTURE-SPECIFIC INFORMATION
  - SUSE LINUX Enterprise Server: START-UP GUIDE
- z/VM documentation - start at:

  `http://www.vm.ibm.com/library/`
  - *z/VM V5R3.0 CP Programming Services*, SC24-6084
  - *z/VM Guide for Automated Installation and Service: Version 5 Release 3.0*, GC24-6099
  - *z/VM CP Messages and Codes: Version 5 Release 3.0*, GC24-6119
  - *z/VM TCP/IP Messages and Codes: Version 5 Release 3.0*, GC24-6124
  - *The Program Directory for Performance Toolkit for VM*, GI10-0785
  - *z/VM CP Commands and Utilities Reference: Version 5 Release 3.0*, SC24-6081
  - *z/VM CP Planning and Administration: Version 5 Release 3.0*, SC24-6083
  - *z/VM Getting Started with Linux on System z9 and zSeries: Version 5 Release 3.0*, SC24-6096
  - *z/VM TCP/IP Planning and Customization: Version 5 Release 3.0*, SC24-6125
  - *z/VM Performance Toolkit Guide*, SC24-6156
  - *z/VM Performance Toolkit Reference*, SC24-6157

© Copyright IBM Corp. 2008. All rights reserved.

– *z/VM V5R3 Guide for Automated Installation and Service*, GC24-6099
– *VMSES/E Introduction and Reference*, GC24-6130

Online resources

These Web sites are also relevant as further information sources:

- The Linux for zSeries and S/390 portal:

  http://linuxvm.org/

- The linux-390 list server:

  http://www2.marist.edu/htbin/wlvindex?linux-390

- Linux on System z and S/390 developerWorks:

  http://awlinux1.alphaworks.ibm.com/developerworks/linux390/index.shtml

- SUSE LINUX Enterprise Server 9 evaluation:

  http://www.novell.com/products/linuxenterpriseserver/eval.html

- z/VM publications:

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

- z/VM performance tips:

  http://www.vm.ibm.com/perf/tips/

- The *z/VM Security and Integrity* paper discusses the isolation and integrity of virtual servers under z/VM

  http://www-1.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130145.pdf

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Index

Symbols

$ALLOC$, z/VM user ID 55, 93, 118, 145, 156
/backup/ directory 153
/etc/fstab 184
/etc/samba/smb.conf file 170
/etc/samba/smbpasswd file 171
/etc/zipl.conf 113, 140
/proc/sys/appldata/ directory 216
/usr/share/doc/ directory 170

Numerics

3270 emulators 25, 41
520RES 35

A

Aduva Onstage 3
Albert Einstein 3
Alt-Tab sequence 29
Apache 162
Apache resources 163
appldata_mem module 216
appldata_net_sum module 216
appldata_os module 216
appletviewer command 176
application development resources 177
application development server 173
APPLMON option 215
ATTACH command 33
Attachmate Extra! 25
AUTOLOG1, z/VM user ID 220

B

Bill of materials 8

C

C/C++ compiler 175
C/C++ development 175
CHANGEPW XEDIT 61
CHANGEPW XEDIT macro 62
Changes to current book xiii
chkconfig command 89, 111, 136, 171
choices made
  in the book 3
CHPW52 XEDIT 61
  source code 228
clone.sh script 153, 170, 174
cloning
  Six more virtual servers 155
cmsfs package 111
cmsfscat command 112
cmsfslst command 112
Common UNIX Printing System (CUPS) 169
Computer Associates
  VM Direct 3
  VM Monitor 201
  VM Secure 61
Control Program 2
Controller xiv
  applying service 111
  Configuration 108
  Copying files to 108
  Installing 97
  Turn off unneeded services 110
  Verify your changes 115
conventions
  command retrieval 10
  in the book 9
  naming backup files 10
  volume labeling 9
CP 2
CPACCESS command 43
CPFMTXA command 48
CPFORMAT EXEC 48
  using it 49
CPSYNTAX 53
CPSYNTAX command 43
CUPS 173

D

DASD
  3390 3
  activating in a SLES10 install 122
  ECKD 3
  formatting in a SLES10 install 123
  labelling convention 9
dasd= parameter
  in zipl.conf 113
dasdfmt command
  using in a for loop 181
dd command 153
desktop 19
df command 141
DIRECTXA command 56, 62, 93, 118, 156
DirMaint 3
Disconnect Timeout 42
Disconnect_Timeout 43
DISKMAP command 54
distributions
  31-bit and 64-bit xiii
Dynamic Memory Upgrade (DMU) 2

E

egrep command 170
envelope files 74
ext3 file system
  creating 184

© Copyright IBM Corp. 2008. All rights reserved.

F

fdasd command 181
Features statement 42
FILELIST command 95
FLASHCOPY command 93, 153
font conventions xv
FORMAT command 56
formatting DASD fast 48
FTP
  site fix 80 subcommand 95
FTP server 46

G

gcc 175
gdb 175
GNU compiler collection (gcc) 175
GNU debugger (gdb) 175
Golden image
  Configuration 132
  Verifying changes 141
golden image xiii

H

hardware resources 8
HELP command 204
HMC 28
  CD-ROM/DVD 30
  Load icon 34
  Operating System Messages 47
  Single Object Operations 29
  SYSG 35
  System Console (SYSC) 47
Hummingbird Host Explorer 25
hypervisor 2

I

IBM Personal Communications 25
IBM RACF 61
IBM Tivoli Provisioning Manager 3
IBM z/VM Performance Toolkit 201
  monitor subcommand 212
INDICATE command 202
INDICATE LOAD command 202
INDICATE QUEUES EXP command 203
infrastructure
  block diagram 4
  design 4
initial RAMdisk 95
install source 109
INSTPLAN 32
INSTVM EXEC 37
Integrated 3270 Console 28, 31
IPL CMS 37
IPWIZARD command 38

J

jar command 176
Java 176
Java development
  tool 176
Java Runtime Environment (JRE) 176
jdb command 177

K

kernel 95

L

LABEL520 EXEC
  source code 231
LAMP 173
LDAP client 167
LDAP server
  manual install 163
ldapsearch command 167
Lightweight Directory Access Protocol (LDAP) 163
Linux
  manual installation 4
  monitoring 201, 215
  monitoring from the kernel 215
Linux distribution 163, 173, 175
Linux PC 85
Linux RMF PM Data Gatherer (rmfpms) 214
LINUX01 145, 162
LNXDFLT profile 92
LNXMAINT 54
  copying files to 58
Load icon 34
logical partition xi, 1, 254
logical volume 93
  adding 180
LPAR
  requirements 8
lsmod command 216
lvcreate command 183
LVM
  block diagram 182
  physical extents 183

M

Mainstar Provisioning Expert 3
MAINT 2CC 220
MAINT CF1 220
make command 176
Map Network Drive 172
Master image xiv
  Installation 120
  Turn off unneeded services 135
MDISK statement 54
Minidisk cache (MDC) 203
Minidisk passwords 61
mke2fs command 184
modprobe command 216
monitor stream support 215
monitoring Linux 214–215
mount command 107

MySQL 173

N

Network File System (NFS) 85
Network Time Protocol (NTP) 192
networking resources 9
NFS server
  configuration on Linux PC 88
nmbd daemon 171
NTP
  setting clock accurately 192
ntpdate command 193

O

OpenLDAP
  pointing virtual server to LDAP server 167
overlaps
  searching for 55

P

paging space 48
paging volumes 47
  location 49
parameter file 95
  copying 156
  copying for new user ID 147
passwords
  planning 12
Performance Toolkit 206
PERFSVM user ID 208
philosophy of the book 3
PHP 173
physical extents 183
planning 7
Printing
  configuration 173
PROFILE EXEC
  AUTOLOG1 59, 94, 119, 146, 220
  LNXMAINT 57
  PERFSVM 212
PROFILE LNXDFLT 92
PROFILE TCPIP 45
Programming 71
PTF 72
PUT2PROD command 37
PuTTY 20
  customizing 20
  larger screen 22
pvcreate command 182
pvdisplay command 182
Python 173–174

Q

QUERY ALLOC MAP command 205
QUERY CPLEVEL command 205
QUERY DASD command 205
QUERY NAMES command 204
QUERY OSA command 158, 205
QUERY PROCESSORS command 205
QUERY STORAGE command 204
QUERY VIRTUAL ALL command 205
QUERY VIRTUAL command 120
QUERY VIRTUAL OSA command 120
QUERY VIRTUAL STORAGE command 121
QUERY VSWITCH ACCESSLIST command 157
QUERY XSTOR command 204
Quick3270 25

R

RAMdisk 95
rcapache2 command 162
rcnfsserver command 115
rcnmb command 171
Reader, z/VM 94
reboot command 114, 141
Recommended Service Upgrade (RSU) 71
Redbooks Web site 244
  Contact us xv
RELEASE command 43
rmfpms 214
rmfpms command 215
root password 130

S

Samba 169
  configuration 170
  integrating with LDAP 171
saved segments
  rebuilding 68
SCSI/FCP disks 3
SET SIGNAL command 59
SET SRM command 59
SET VSWITCH GRANT command 94, 120, 146, 157
share /usr/ file system 3
SHUTDOWN command 47
SHUTDOWN REIPL command 53, 60
Single Object Operations 29
site fix 80, FTP subcommand 95
SLES 10 SP2
  install source 109
SLES10 91
  activating DASD in an install 122
  applying service 136
  bootstrap files 94
  formatting DASD in an install 123
  installation 121
  Installing and configuring 91, 117
  Java 176
  parameter file 147
  partitioning DASD in an install 123
  Preparing for Online Update 135
  second part of an install 129
SLES10 EXEC 96, 121
smbd daemon 171
smbpasswd command 171
software resources 8
SSH 132

Archived

Configuring keys 139
key-based authentication 139
protocol 2 20
Standalone Program Loader 35
Standard Development Kit (SDK) 176
starter system 121
Storage Area Network (SAN) 3
SuSE Linux Enterprise Server xiii
SUSE Linux Enterprise Server (SLES) xiii, 94, 96
swapon command 107
SYSG 35
system administration
roles 12
system block diagram
LPAR and user ID based 4
SYSTEM CONFIG 43, 64, 220
customizing 41
SYSTEM CONFIG file
adding PAGE and PERM disks 52
Features statement 42
System Console (SYSC) 47
SYSTEM DTCPARMS 220
system shutdown 59
system startup 59
System status
DASD view 158
System z
virtualization capabilities 2
System z10 2
System_Identifier_Default 42

T

tar command 214
TCL 174
TCP/IP
z/VM configuration 38
TCP/IP resources 9
TCPCMSU 54
TCPIP Data 220
TCPMAINT 198 220
TCPMAINT 592 220
Tool Command Language (TCL) 174

U

UM97510 77
UM97520 77
UM97530 77
Universal Naming Convention (UNC) 172
usability tests
on the book 5
USER DIRECT 220
USER DIRECT file 54, 92, 118, 155
APPLMON option 215
INCLUDE statement 92
profile 92
USER DISKMAP file 55
User_Volume_Include 52, 145
User_Volume_List 52
userid() function 96

V

VDISK 42
Vdisk
allowing creation of 43
Velocity Software ESALPS 201
vgcreate command 183
virtual disk (VDISK) 42
virtual NIC 93
Virtual server
application development 173
File and print 169
LDAP 163
Web server 162
virtual servers 2
virtual switch (VSWITCH) 1, 42
Virtualization 1
vmcp command 115
VMLINK command 50
VNC 121
configuring on master image 132
volume group 183
VSWITCH 1, 44, 47, 157

W

What is virtualization 2
which command 153
Windows workstation 85
worksheets
examples in the book 13

X

XAUTOLOG command 44, 59, 153
XEDIT
"" prefix command 92, 155
? subcommand 10
add subcommand 43
BOTTOM subcommand 43
CHANGE subcommand 62
FILE subcommand 43
PREFIX OFF subcommand 55
search subcommand (/) 42
XEDIT command 41
XEDIT PROFILE 41

Y

yast
Setting LDAP client 190

Z

z/VM
adding paging volumes 47
backing up to tape 62
cons=sysg 35
conventions 9
CPFMTXA command 48
customizing shutdown 59
customizing startup 59

Archived

DIRECTXA command 56
Disconnect Timeout 42
DISKMAP command 54
envelope files 74
FORMAT command 56
FTP server 46
HELP command 204
HIT RATIO 203
INDICATE command 202
INDICATE QUEUES EXP command 203
INSTPLAN EXEC 32
INSTVM EXEC 37
Integrated 3270 Console 28
IPWIZARD command 38
latest PTFs 77
MIGRATE rate 203
monitoring 201
paging efficiency 49
PAGING rate 203
PUT2PROD command 37
QUERY ALLOC MAP command 205
QUERY CPLEVEL command 205
QUERY DASD command 205
QUERY NAMES command 204
QUERY OSA command 205
QUERY PROCESSORS command 205
QUERY STORAGE command 204
QUERY VIRTUAL ALL command 205
QUERY XSTOR command 204
reader 94
relabeling system volumes 63
RELEASE command 43
Restoring from tape 68
security issues 60
Servicing 71
SET SIGNAL command 59
Standalone Program Loader 35
STEAL value 203
strengths 1
system name 42
TCP/IP configuration 38
TCPIP at IPL time 44
userid() function 96
virtual switch 1
VSWITCH 1
XEDIT PROFILE 41
XSTORE rate 203
z/VM 5.3
improvements 2
z/VM 5.4
improvements 2
z/VM CPACCESS command 43
z/VM system
backup Copy 63
z/VM user IDs
default password 38
zipl.conf 113, 140

Archived

Archived

The Virtualization Cookbook for SLES 10 SP2

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM Confidential

Archived

Archived

z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2

Hands-on instructions for installing z/VM and Linux on the mainframe

Updated information for z/VM 5.4 and Linux SLES 10 SP2

New, more versatile file system layout

This IBM Redbooks publication describes how to roll your own Linux virtual servers on IBM System z hardware under z/VM. This edition applies to Version 5, Release 4, Modification 0 of z/VM (product number 5741-A05) and Novell SUSE Linux Enterprise Server (SLES) Version 10, Service Pack 2 (SP2). 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 deployed in a matter of minutes. This powerful build and clone capability can enable you to launch new products and services without the exhaustive planning, purchasing, installing and configuring of new hardware and software that can be associated with conventional discrete hardware servers. Development groups that need test environments built and rebuilt rapidly to enable them to efficiently deliver their projects, while handling change management in the process, can also benefit from this unique advantage.

The publication adopts a cookbook format that provides you with a concise, repeatable set of procedures for installing and configuring z/VM in a logical partition (LPAR), and then installing and customizing Linux. To implement these instructions, you need an IBM System z logical partition with associated resources, z/VM 5.4 media, and a Linux distribution.

This book assumes that you have a general familiarity with System z technology and terminology. It does not assume an in-depth understanding of z/VM and Linux. It is written for those who want to get a quick start with z/VM and Linux on the mainframe.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7493-00 ISBN 0738431826