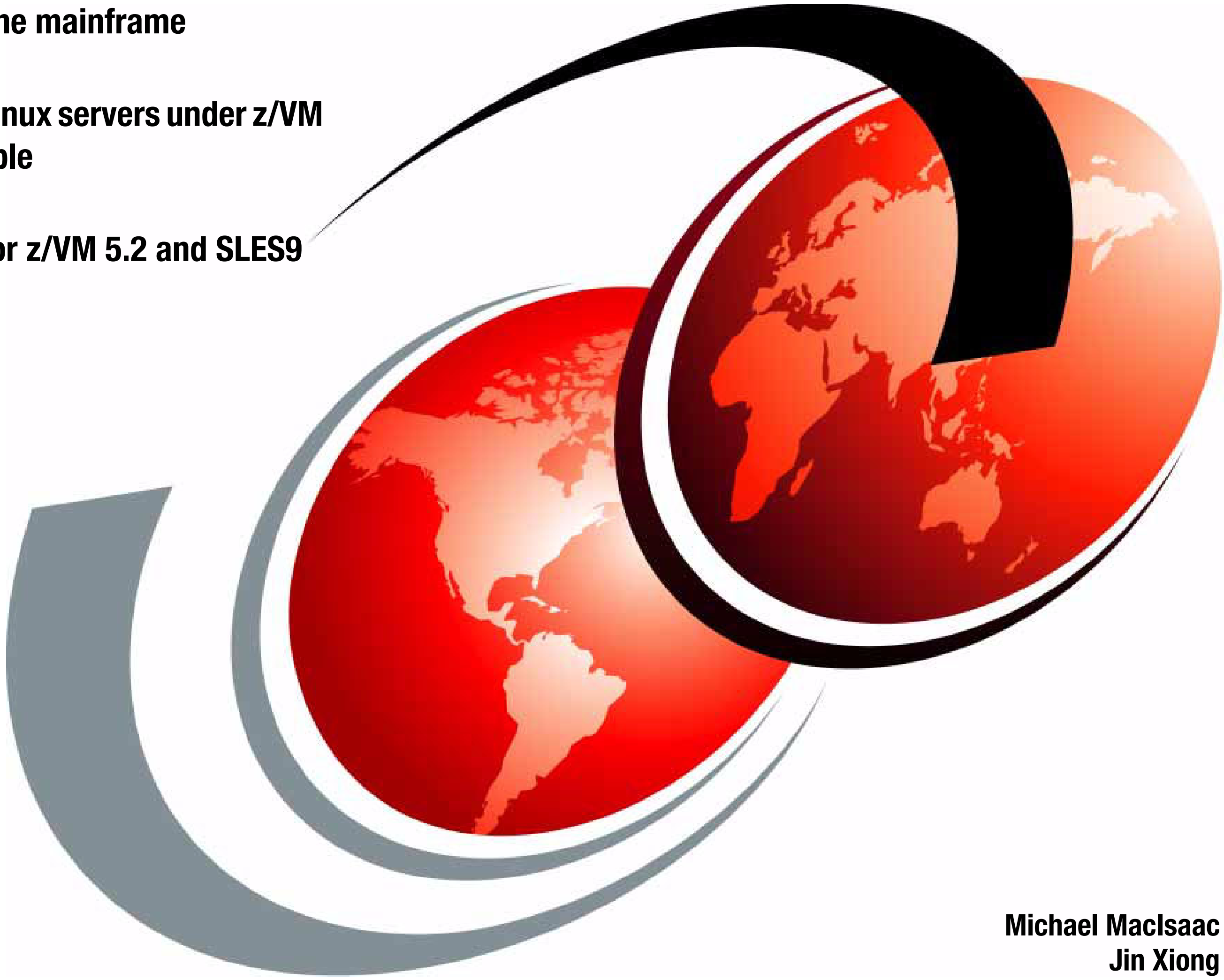
z/VM and Linux on IBM System z:

The Virtualization Cookbook for SLES9

A cookbook for installing z/VM and Linux on the mainframe

Running Linux servers under z/VM made simple

Updated for z/VM 5.2 and SLES9 SP3



Michael MacIsaac
Jin Xiong

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9

April 2006

SG24-6695-01

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

Second Edition (April 2006)

This edition applies to z/VM Version 5, Release 2 and multiple Linux distributions. SuSE LINUX Enterprise Server 9, Service Pack 3 is used for the examples in this book.

Contents

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

AIX®
DirMaint™
Domino®
ECKD™
eserver®
eserver®
eServer™
HiperSockets™
IBM®
iSeries™
Lotus®
RACF®
Redbooks (logo) ™
Redbooks™
RMF™
S/390®
System z9™
Tivoli®
Workplace™
z/OS®
z/VM®
z9™
zSeries®

The following terms are trademarks of other companies:

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

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

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, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook describes how to setup your own Linux® virtual servers on IBM zSeries® and System z9™ under z/VM®. It adopts a cookbook format that provides a clearly documented set of procedures for installing and configuring z/VM in an LPAR and then installing and customizing Linux. You need a zSeries logical partition (LPAR) with associated resources, z/VM 5.2 media, and a Linux distribution. This book is based on SUSE Linux Enterprise Server 9 (SLES9) for zSeries and we address both 31-bit and 64-bit distributions.

In addition, there are a few associated REXX EXECs and Linux scripts to help speed up the process. These tools are not IBM products nor formally supported. However, they are informally supported. They are available on the Web.

In this book, we assume that you have a general familiarity with zSeries technology and terminology. We do not assume an in-depth understanding of z/VM and Linux. This book 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

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

Jin Xiong tests IBM server technologies and Linux on IBM zSeries in Poughkeepsie, NY.

This book was originally written in 2005 by Michael MacIsaac, Jin Xiong, and Curtis Gearhart. It was updated in 2006 for z/VM 5.2 and SLES-9 SP3 by Michael MacIsaac and Jin Xiong. The DCSS/XIP2 section was based on work by Carlos Ordonez, which was, in turn, based on work by Carsten Otte. We sincerely thank the following people who contributed to this project in various ways:

David Boyes, Adam Thornton
Sine Nomine Associates

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

Duane Beyer, Michel Considine, Roy Costa, Greg Geiselhart, Tony Giaccone, Dionne Graff, Susan Greenlee, Bill Norton, Carlos Ordonez, Bruce Smilowitz, Kyle Smith, Helen Tsang, Donna Von Dehsen, Hossee Wakil and Dennis Wunder
IBM Poughkeepsie

Rob van der Heij
Velocity Software

Carsten Otte, Claudia Prawirakusumah
IBM Germany

Chester Hood
State of Tennessee

Jerry Epker
Fidelity Information Services

Bernard Wu
NCCI

Thanks the zBMC team in IBM Poughkeepsie and to the many who answered questions on the linux-390 list server.

Conventions

The following font conventions are used in this book:

`Monospace and bold`	Commands entered by the user on the command line
`<value>`	Value inside angle brackets is to be replaced
`monospace`	File, directory and user ID 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 that will not fit on one line are suffixed with \
- Linux commands running as non-root are prefixed with **$**

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll 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 Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

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

2455 South Road
Poughkeepsie, NY 12601-5400

Summary of changes

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

April 2006, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

This book was first published in June of 2005 as an IBM redbook entitled *z/VM and Linux on zSeries: From LPAR to Virtual Servers in Two Days*. The new title, *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9*, hopefully better reflects the contents of the book and the mainframe's new brand name.

Some content has been removed, some added and some changed.

What has been removed

The following chapters and sections have been removed, largely because there was not a lot of interest shown and insufficient resource to properly test and update them:

- Chapter 10: *Virtual communications server*
- Chapter 11: *Virtual communications controller server*
- Section 9.5: *An Eclipse application development system*
- Appendix A: *The SYSVINIT package*

Also, the description of creating two VSWITCH controllers has been removed because it no longer necessary. Two user IDs that perform this function, `DTCVSW1` and `DTCVSW2`, are now standard with z/VM 5.2

What has been added

The following chapter and sections have been added:

- A description of setting up a VNC client on a Windows® desktop and the VNC server on zSeries Linux.
- A new Chapter 5, "Servicing z/VM" on page 63 that describes how to apply a Programming Temporary Fix (PTF) and a Recommended Service Upgrade (RSU).
- A basic description of how to add a second controller (section 8.4, "Creating a SLES9 31-bit controller" on page 128): SLES-9 is a release that allows a transition from 31-bit (s390) architecture to 64-bit (s390x). Often enterprises will require some servers of each type. Utilizing a second controller will allow you to clone either 31-bit (s390) or 64-bit (s390x) servers.
- A recipe for a DCSS/XIP2 shared file system (section 11.2, "Creating a DCSS/XIP2 shared file system" on page 175)
- A recipe to create a logical volume (section 11.1, "Adding a logical volume" on page 168): Often more disk space is needed on a virtual server so an example has been added.

What has been changed

The following aspects of the book have changed.

- The term *master image* is used rather than *golden image*. This is the copy of Linux that is to be cloned.
- Less magic, more description and fewer associated files: The first edition of the book often used scripts to help save time. From interaction with readers doing hands-on, it was learned that they would often not want to use these scripts. They would rather spend more time doing the tasks manually to learn and understand their system better, even if this requires more time to get the system set up. Therefore, many of the associated files have been removed.
- The **`vmcp`** module/command has replaced the cpint package/**`hcp`** command by Neale Ferguson. Either tool will perform the job of sending commands from Linux to z/VM's CP adequately. While cpint is shipped with SLES9, it was never picked up by Red Hat. RHEL4 Update 3 has incorporated **`vmcp`** so using it will make the cloning process more portable. However, cpint is still described and a copy of the clone script that uses it is included for readers who will not be installing the latest distribution.
- The **`FLASHCOPY`** call has been removed from `CPFORMAT EXEC`. It was assumed that **`FLACHCOPY`**ing a DASD that had been formatted with **`CPFMTXA`** to a new volume would both save time and function properly. While it did save time, it did not appear to function properly on volumes that have *never* been formatted.
- The **`mksles9root.sh`** script has been updated for SLES9 SP3 and to work around an apparent bug in the YaST *Patch Update CD* function.
- The **`clone.sh`** script has been updated so it does not require any interaction after the initial "Are you sure" confirmation. This change includes removing the setting of the root password, so all clones will have the same root password. You can change the root password immediately after it has been cloned. The logic has been changed so that the master image is never modified, rather, it is cloned and the target minidisks are modified.

1

Introduction to z/VM and Linux

Virtualization is hot! However, mainframe and VM, now z/VM, have had virtualization for decades. When Linux came to the mainframe in 2000, it was a natural fit to run under z/VM. You can run tens of Linux images on the same zSeries logical partition (LPAR).

With a z/VM and Linux infrastructure, you can reduce the time between installing new servers and implementing them because new servers can be deployed in less than an hour. With this powerful capability, you can launch new products and services without planning, purchasing, installing and configuring new hardware and software. Development groups who need test environments rapidly to handle change management efficiently can also benefit from this unique advantage. Some of mainframe's and z/VM's best strengths are:

- z/VM and the mainframe's virtualization capabilities are more mature and robust than any other hardware and virtualization combination.
- z/VM's virtual switch (VSWITCH) has made Linux networking 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. There are only a handful of configuration files. Set it up once, and z/VM runs for months with little maintenance and administration required.

z/VM 5.2, available as of December of 2005, provides major improvements when operating on z9-109 or zSeries servers with large memory configurations. Scalability is improved 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 done 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, memory-intensive, virtual server environments. Read more about zSeries virtualization capabilities on the Web:

`http://www-1.ibm.com/servers/eserver/zseries/virtualization/features.html`

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 (storage), network adapters (OSA cards) and CPU (CPs or IFLs). These resources are managed by a *hypervisor*. z/VM's hypervisor is called Control Program (CP). When the user logs onto z/VM, the hypervisor creates a virtual machine which can run one of many different mainframe operating systems. The two operating systems that are discussed in this book are the Conversational Monitoring System (CMS) and Linux. CMS can be thought of as a z/VM shell. Virtual machines running Linux become the virtual servers.

1.2 A philosophy adopted in this book

An important philosophy adopted in this book is to keep all solutions simple. Two common expressions used are "the KISS method" (Keep It Simple, Stupid) and the quote from Albert Einstein at the start of this chapter: *Everything should be made as simple as possible, but not simpler*. Because KISS is somewhat condescending and because 2005 was both the first year this book was published and the centennial anniversary of Albert Einstein's many famous papers, this book uses his ideas.

A lot of books and papers are talking about virtualization today, but not telling you how to do it. This book puts the *how to* behind these marketing words.

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 more important choices and assumptions made in writing this book:

- Cloning product versus customized cloning

  Cloning products, such as Aduva's Onstage, IBM Tivoli® Provisioning Manager, IBM Director function z/VM Center (briefly discussed in 1.4, "IBM Director and z/VM Center Extension" on page 3) and Levanta, are outside the scope of this book. While these are all viable solutions, the cloning described in this book allows you to customize Linux images without requiring such products. However, these products are more sophisticated than the simple EXECs and scripts in this book.

- Directory Maintenance product versus the `USER DIRECT` file

  The `USER DIRECT` file is chosen over a directory maintenance product such as IBM *DirMaint™* or Computer Associates' *VM:Direct*. If you feel that DirMaint as a directory maintenance product is better for your enterprise, use the book *Getting Started With Linux*, SC24-6096, to configure z/VM. You can still use this book to configure Linux.

- Provisioning versus predefined user IDs

  z/VM user IDs must be predefined to clone. There is no attempt to *provision* (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 or the cloning script will fail.

- Shared read-only Linux `/usr/` file system versus read/write

  Many cloning solutions use an environment which shares the `/usr/` file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. A read/write `/usr/` file system on the virtual servers is chosen to keep things as simple as possible. However, utilizing a z/VM DCSS to share executable files is discussed.

- Conventional 3390 ECKD™ direct access storage device (DASD) versus FBA disks accessed via SCSI over FCP

  The zSeries server has traditionally only supported 3390 (or older 3380) DASD. Support has been extended to include SCSI/FBA disks in storage area networks (SANs). The support of FBA disks is more complicated than conventional DASD. In keeping things as simple as possible, only conventional DASD is described in this book.

- Cloning script or EXEC versus manual installation

  It is easy to spend more time setting up an infrastructure for cloning Linux under z/VM than the time that it saves you over manually installing Linux, given the number of times you actually clone. When cloning works quickly, it can be an extremely useful tool in your toolbox. Therefore, this book discusses both cloning and manual installation. These two methods of provisioning Linux servers hinge on the existence of an installation parameter file for each Linux user ID. If you want a more complete solution, the products recommended in the first bullet point are recommended.

1.4 IBM Director and z/VM Center Extension

IBM Director 5.10 brings a comprehensive management functionality to Linux on IBM system z. The base IBM Director functions (e.g. monitoring, event action plans, software distribution, inventory, remote control, task scheduling) are now provided for any Linux end point on System z. In addition, the z/VM Center Extension provides further functionality for provisioning and configuration of z/VM Linux guests.

The z/VM Center extension includes the following tasks:

- Virtual Server Deployment - creation of virtual servers and deployment of operating systems into them by using virtual server and operating system templates, management of virtual servers (create/delete/activate/properties) and provisioning resources.
- Server Complexes - automatic fashion of controlling the configuration (and creation) of groups of Linux guests, handling both the z/VM side and Linux side aspects, supporting z/VM Resource Manager performance goals, virtual networking (based on VM Guest LAN, OSA and VSWITCH), z/VM minidisk attachments and configuration scripts.

The integration of the z/VM Center virtualization functionality with the full breadth of IBM Director on Linux managed end points provides a powerful tool for managing Linux guest colonies on z/VM systems.

1.5 Infrastructure design

To clone Linux, or *provision virtual servers*, there must be a certain infrastructure design in place. A zSeries server with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-1 on page 4 shows a block diagram of a z990 with many LPARs. z/VM 5.2 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. Important z/VM minidisks and configuration files are shown next to each user ID.

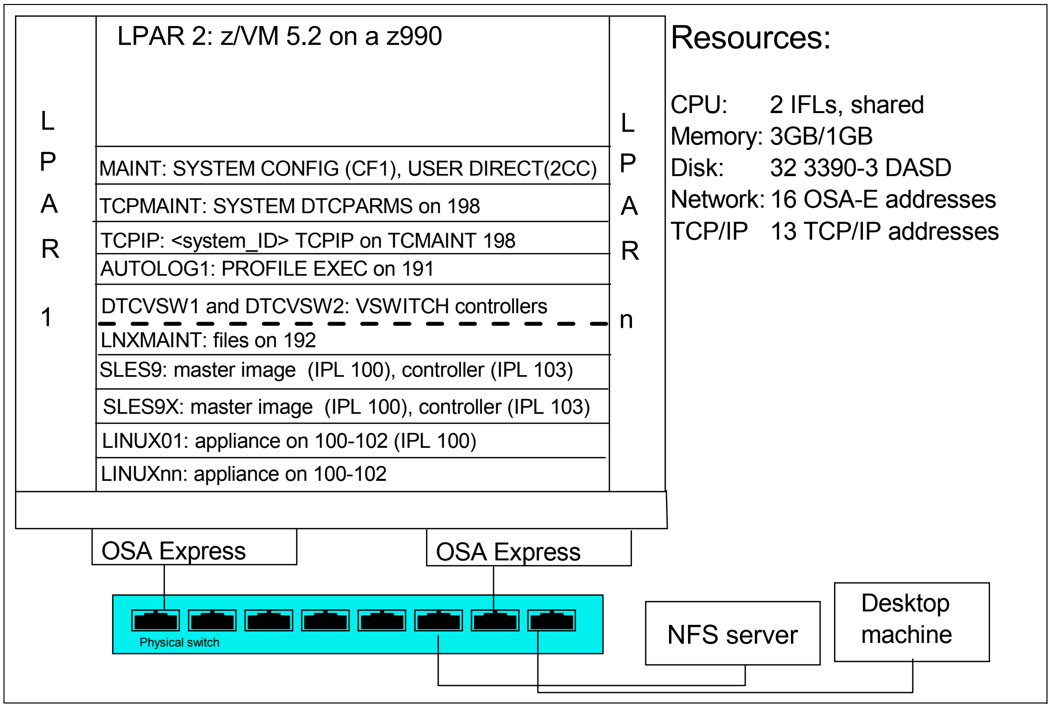

LPAR 2: z/VM 5.2 on a z990
L P A R 1
L P A R n
MAINT: SYSTEM CONFIG (CF1), USER DIRECT(2CC)
TCPMAINT: SYSTEM DTCPARMS on 198
TCPIP: <system_ID> TCPIP on TCMAINT 198
AUTOLOG1: PROFILE EXEC on 191
DTCVSW1 and DTCVSW2: VSWITCH controllers
LNXMAINT: files on 192
SLES9: master image (IPL 100), controller (IPL 103)
SLES9X: master image (IPL 100), controller (IPL 103)
LINUX01: appliance on 100-102 (IPL 100)
LINUXnn: appliance on 100-102
Resources:
CPU: 2 IFLs, shared
Memory: 3GB/1GB
Disk: 32 3390-3 DASD
Network: 16 OSA-E addresses
TCP/IP 13 TCP/IP addresses
OSA Express
OSA Express
Physical switch
NFS server
Desktop machine

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

The user IDs above the dashed line are those important user IDs defined in z/VM 5.2. The user IDs below the dashed line are described in this book and have the following functions:

- LNXMAINT: A user ID on which to store files that will be used by both CMS and Linux
- SLES9: The 31-bit master Linux image (the copy of Linux to be cloned) and the controller Linux image (the Linux image doing the cloning and other services)
- SLES9X: The 64-bit master Linux image and the controller Linux image
- LINUXnn: The Linux virtual servers, `LINUX02-LINUX07`, where Linux can either be cloned to or installed manually, each virtual server is configured with a single 3390-3 minidisk which is slightly more than 2 GB of space.

1.6 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, the participants had all completed up to Chapter 6, “Configuring an NFS server” on page 75, so z/VM was installed and customized for TCP/IP communications with a highly available VSWITCH. By the end of the second day, all participants had cloned their first Linux virtual server.

You should be able to complete most of the virtual servers described in the book with three days.

1.7 The chapters in this book

The remaining chapters and appendixes in this book are summarized in the following list:

- 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 a password planning. Sample worksheets are provided for the examples used in the book, as are blank copies for your use.
- Chapter 3, “Configuring a desktop machine” on page 17 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 25 shows how to install and configure z/VM. This is where you roll up your sleeves and start to work.
- Chapter 5, “Servicing z/VM” on page 63 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” on page 75, 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 zSeries controller Linux is installed, you can copy the Linux install tree to it and retire the Linux PC server.
- Chapter 7, “Installing and configuring Linux” on page 85, describes how to install and configure two Linux images onto the first Linux user ID: the *master image*, which it is cloned from, and the *controller*, which does the cloning among other tasks.
- Chapter 8, “Configuring NFS on controller” on page 123, illustrates how to move the Linux install tree from the Linux PC server to the controller under z/VM.
- Chapter 9, “Configure Linux for cloning” on page 131 explains how to prepare z/VM user IDs and clone your first virtual server.
- Chapter 10, “Four virtual servers” on page 145, 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 167 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 187, describes basic steps to begin monitoring z/VM and your new Linux virtual servers.
- Chapter 13, “Backup and restore” on page 203, shows basic steps on how to back up these new systems.
- Appendix A, “References and source code” on page 209, provides information about the files on the CD or associated tar file.

2

Planning

This chapter covers the planning that should be done before installing z/VM. It begins by discussing a *bill of materials*, or all the resources that you need. Then it explains the labeling of 3390 volumes. Finally resource worksheets are presented for:

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

2.1 Bill of materials

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

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

- A zSeries logical partition (LPAR); z800, z900, z890 or z990, or System z9
  - Processors or CPUs: One IFL minimum, two or more recommended
  - Memory: 3 GB central/1 GB expanded minimum, 6 GB/2 GB or more recommended. This 3:1 ratio of central to expanded storage is a good starting point. See the following Web site for a discussion of how to apportion memory:

    `http://www.vm.ibm.com/perf/tips/storconf.html`
  - DASD: 32 3390-3s minimum to start (It is helpful to have DASD on different CHPIDs and in different host bay adapters for better z/VM paging performance.) If you cannot get 32, then 24 should be the very minimum.
  - Open Systems Adapter (OSA) network cards: One card minimum with 12 device numbers. Two cards with eight device numbers on one and four on the other is recommended for high availability.
- A computer that will act as an Network File System (NFS) server temporarily with at least 12 GB of disk space (Linux PC or UNIX® server is recommended) and connected to the network.
- A workstation or desktop that has network access to the mainframe

2.1.2 Software resources

The following software resources are needed:

- z/VM 5.2 install media with documentation (DVD install is described in this book.)
- Linux install media (SLES9 SP3 is described in this book.)
- An operating system for the NFS server (SLES9 and RHEL4 are described in this book.)
- The code associated with this book, on the Web:

  `ftp://www.redbooks.ibm.com/redbooks/sg246695`
- Tools on the workstation and desktop:
  - A 3270 Emulator such as *Attachmate Extra*, Hummingbird *Host Explorer*, or IBM *Personal Communications* for Windows desktops (for Linux desktops, a 3270 emulator named *x3270* is available)
  - A Linux SSH client such as PuTTY (recommended) or TeraTerm (for Linux desktops the **ssh** client is built in)

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 should be routed to the OSA card(s).

2.2 z/VM conventions

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

2.2.1 Volume labeling convention

You should have a convention for labeling DASD. Your shop 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 zSeries DASD is addressed with a device number consisting of four hexadecimal digits. Each zSeries 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 DASD 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 *owned* by other LPARs. In this situation, it is convenient to identify the LPAR that *owns* the DASD. Therefore the volume labeling convention used in this book identifies the LPAR via 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 *V*. 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

> **Note:** The labels are 520RES, 520W01, 520W02, 520SPL, and 520PAG when z/VM is installed.

For example, Figure 2-1 shows the labeling convention for the DASD in LPAR *V*, of type *minidisk* at real address *E34A*.

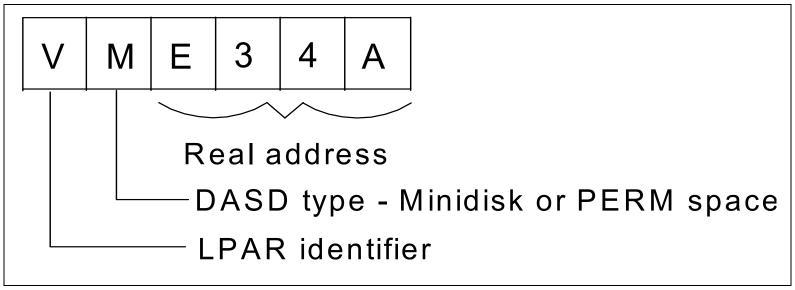


Figure 2-1 DASD labeling convention

The letter V is hard-coded into two 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 recommend 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. Since 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 USER DIREORIG file.

2.2.3 The command retrieve convention

The ability to retrieve past commands is a common tool. Often it is nice to retrieve in both directions in case you "pass" the command you're 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, though 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* since it is right next to F12. Also, the same function is useful in the editor, **`XEDIT`**. The **`?`** subcommand retrieves past commands, so it is recommended that you assign it to **`F12`**.

2.3 Password planning

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

This book considers system administration roles:

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

The z/VM and Linux system administrator may well 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.

Because the passwords to the z/VM Linux user IDs are the same as the system IDs, such as MAINT, this assumes that the Linux end users will not log onto z/VM 3270 sessions. The root passwords of the cloned Linux virtual servers will be different, so the Linux virtual server end users will not inherit the root password of the Linux master image.

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 end user (with or without authority for 3270 sessions)

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

2.4 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 2.5, "Blank worksheets" on page 14.

2.4.1 z/VM resources used in this book

Table 2-1 lists the z/VM resource values used in the examples in this book.

Table 2-1 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	LNXVM52	Name to be assigned to z/VM system
TCP/IP host name	lat124	Assigned by a network administrator; helpful to set in DNS beforehand, but not necessary
TCP/IP domain name	pbm.ihost.com	Helpful to set in DNS beforehand
TCP/IP gateway	129.40.178.254	The router to and from the local subnet
DNS server 1	129.40.106.1	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	3000	Start of OSA *triplet* for the z/VM TCP/IP stack
TCP/IP address	129.40.178.124	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

Name	Value	Comment
Network type	Ethernet	Usually “Ethernet”
Port name (optional)		Not required by z/VM
Router type	None	Usually “None”
Primary OSA device number for VSWITCH	3004	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	3008	Should be on a different CHPID/OSA card

2.4.2 z/VM DASD used in this book

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

Table 2-2 z/VM DASD worksheet

Device number	Label	Type	Notes
A770	520RES	CP owned	System residence volume (520RES by default)
A771	520W01	CP owned	W01 volume (520W01 by default)
A772	520W02	CP owned	W02 volume (520W02 by default)
A773	520SPL	System spool	Spool volume 1 from z/VM installation (520SPL)
A774	520PAG	System paging	System paging volume 1 from z/VM installation (520PAG)
A775	VPA775	System paging	System paging volume 2
A776	VPA776	System paging	System paging volume 3
A777	VPA777	System paging	System paging Volume 4
A778	VPA778	System paging	System paging Volume 5
A779	VPA779	System paging	System paging volume 6
A77A	VMA77A	System minidisk	SLES9 104
A77B	VMA77B	System minidisk	SLES9 100 and 102, the *master image*
A77C	VMA77C	System minidisk	SLES9 103, the *controller*
A77D	VMA77D	System minidisk	SLES9 105, part of the `/nfs/` logical volume
A77E	VMA77E	System minidisk	SLES9 106, part of the `/nfs/` logical volume
A77F	VMA77F	System minidisk	SLES9 107, part of the `/nfs/` logical volume
A780	VMA780	System minidisk	SLES9 108, part of the `/nfs/` logical volume
A781	VMA781	System minidisk	SLES9X 100 and 102, the *master image*
A782	VMA782	System minidisk	SLES9X 103, the *controller*
A783	VMA783	System minidisk	SLES9X 104 and LNXMAINT 191 and 192
A784	VMA784	System minidisk	SLES9X 105, part of the `/nfs/` logical volume

Device number	Label	Type	Notes
A785	VMA785	System minidisk	SLES9X 106, part of the `/nfs/` logical volume
A786	VMA786	System minidisk	SLES9X 107, part of the `/nfs/` logical volume
A787	VMA787	System minidisk	SLES9X 108, part of the `/nfs/` logical volume
A788	VMA788	System minidisk	LINUX01 100 and 102
A789	VMA789	System minidisk	LINUX02 100 and 102
A78A	VMA78A	System minidisk	LINUX03 100 and 102
A78B	VMA78B	System minidisk	LINUX04 100 and 102
A78C	VMA78C	System minidisk	LINUX05 100 and 102
A78D	VMA78D	System minidisk	LINUX06 100 and 102
A78E	VMA78E	System minidisk	LINUX07 100 and 102

2.4.3 Linux resources used in this book

Table 2-3 lists the Linux resources used in the examples in this book.

Table 2-3 Linux resources worksheet

Name	Value	Comment
Linux install password	lnx4vm	
Linux TCP/IP gateway	129.40.178.254	
Linux TCP/IP broadcast	129.40.178.255	
Linux DNS server	129.40.106.1	Often the same as z/VM's
NFS server TCP/IP address	129.40.46.206	
VNC installation password	lnx4vm	

2.4.4 Linux user IDs used in this book

Table 2-4 lists the Linux user IDs used in the examples in this book.

Table 2-4 Linux user ID worksheet

Linux user ID	IP address	DNS name	Notes
SLES9X	129.40.178.127	lat127.pbm.ihost.com	A 64-bit controller and master image
SLES9	129.40.178.128	lat128.pbm.ihost.com	A 31-bit controller and master image
LINUX01	129.40.178.131	lat131.pbm.ihost.com	A Web virtual server
LINUX02	129.40.178.132	lat132.pbm.ihost.com	An LDAP virtual server
LINUX03	129.40.178.133	lat133.pbm.ihost.com	A file and print virtual server
LINUX04	129.40.178.134	lat44.pbm.ihost.com	A basic application development virtual server
LINUX05	129.40.178.125	lat45.pbm.ihost.com	An extra virtual server

Linux user ID	IP address	DNS name	Notes
LINUX06	129.40.178.126	lat46.pbm.ihost.com	An extra virtual server

2.5 Blank worksheets

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

2.5.1 z/VM resources worksheet

Use the worksheet in Table 2-5 to document the z/VM resources that you will use.

Table 2-5 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		

2.5.2 z/VM DASD worksheet

Use the worksheet in Table 2-6 to document the z/VM DASD that you will use. The examples in this book use twenty 3390-3s (see 2.4.2, “z/VM DASD used in this book” on page 12), five each for the z/VM system, paging space, the LNXMAINT and SLES9 user IDs, and seven for the seven Linux virtual servers.

Table 2-6 z/VM DASD blank worksheet

Device number	Label	Type	Notes

2.5.3 Linux resources worksheet

Use the worksheet in Table 2-7 to document your Linux resources.

Table 2-7 Linux resources blank worksheet

Name	Value	Comment
NFS server TCP/IP address		
Linux install password		
Linux TCP/IP gateway		
Linux TCP/IP broadcast		
Linux DNS server		
VNC Installation password		

2.5.4 Linux user ID worksheet

Use the worksheet in Table 2-8 to document the Linux user IDs that you will create.

Table 2-8 Linux user ID blank worksheet

Linux user ID	IP address	DNS name	Notes

3

Configuring a desktop machine

Many people use Microsoft® Windows as a desktop operating system. This chapter addresses the following tools that 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

3.1 PuTTY: A free SSH client for Windows

Throughout this book, SSH is used to log into Linux systems. It is simple 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 want a good SSH client.

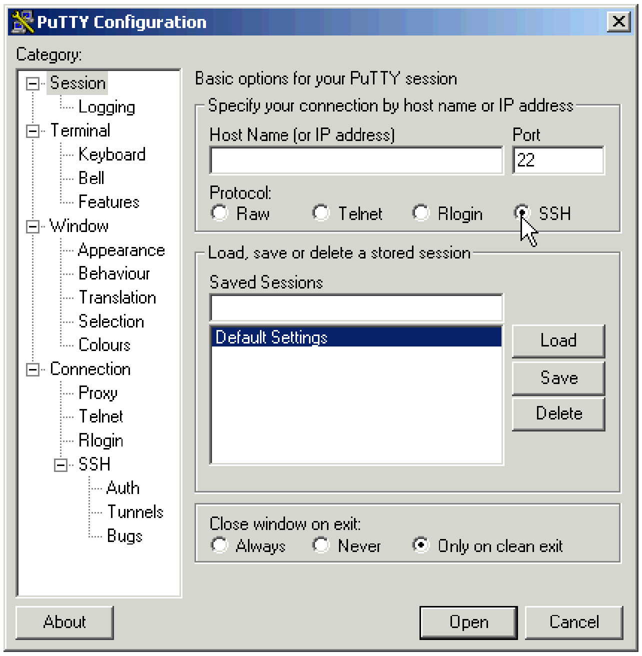
PuTTY is probably the most commonly used. You can also find a PuTTY client for Windows on CD1 of a SUSE LINUX Enterprise Server (SLES) distribution in the `/dosutils/putty` file. Or you can download it from the Web at:

`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`. You might want to include the version in the file name such as `putty-0.56.exe`. PuTTY is a stand-alone executable. There is no formal installation process. You might also want to create a shortcut on your desktop or task bar.

Invoke PuTTY. The PuTTY Configuration window (Figure 3-1) should open. Spend a few minutes to configure PuTTY, which will save you time later.

1. In the PuTTY Configuration window, complete these tasks:
   a. Under the Protocol heading on the right, click the **SSH** radio button.
   b. In the left Category panel, click **SSH**.



Port: 22

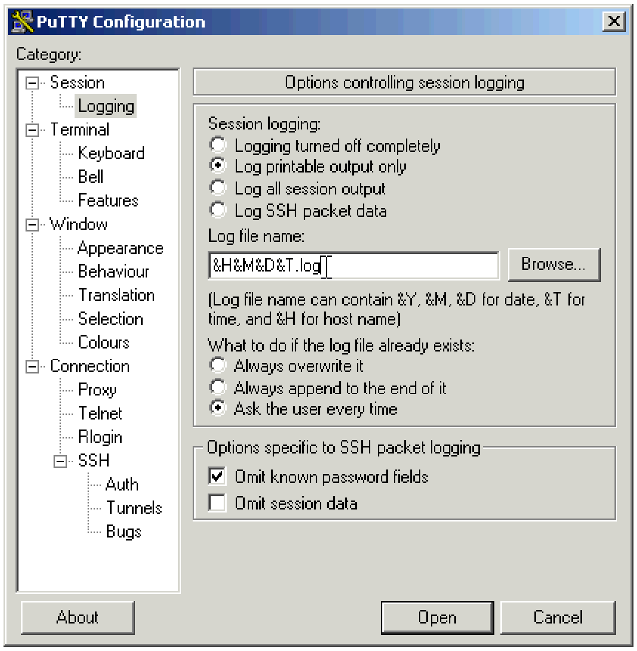
Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 3-1 PuTTY Configuration window

2. Click **Logging (**Figure 3-2 on page 19**)**. Click the radio button **Log printable output only**. By doing this, you can go back and check on the output of certain commands. Set the Log file name to **&H&M&D&T.log** so it will include a time stamp in the file name.



Options controlling session logging

Session logging:

○ Logging turned off completely
● Log printable output only
○ Log all session output
○ Log SSH packet data

Log file name: &H&M&D&T.log

(Log file name can contain &Y, &M, &D for date, &T for time, and &H for host name)

What to do if the log file already exists:

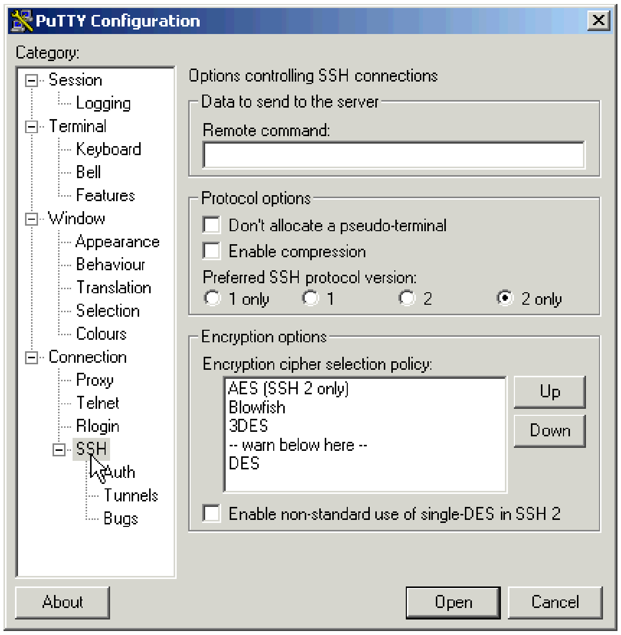
○ Always overwrite it
○ Always append to the end of it
● Ask the user every time

Options specific to SSH packet logging

☑ Omit known password fields
☐ Omit session data

Figure 3-2   Setting Logging

3. The window changes as shown in Figure 3-3. Complete these steps:
   a. On the right side of the window, under Preferred SSH protocol version, click the **2 only** radio button.
   b. In the left Category panel, click **Terminal**.



Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

☐ Don't allocate a pseudo-terminal
☐ Enable compression

Preferred SSH protocol version:
○ 1 only ○ 1 ○ 2 ● 2 only

Encryption options

Encryption cipher selection policy:

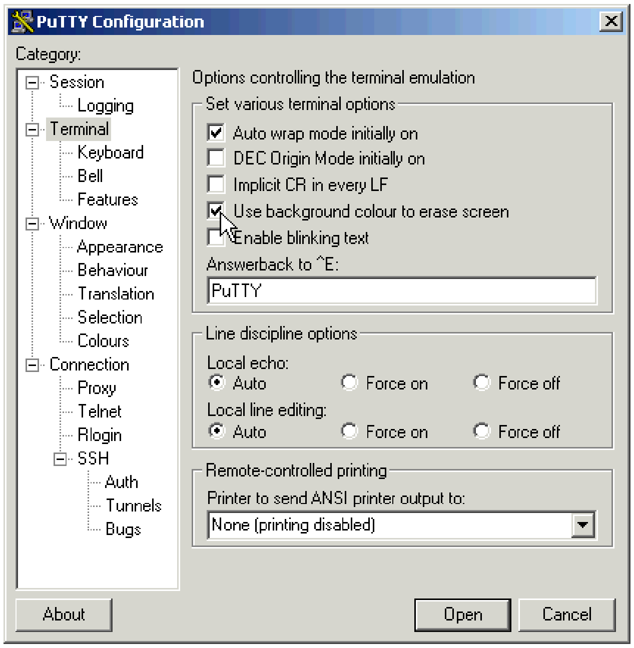
AES (SSH 2 only)
Blowfish
3DES
-- warn below here --
DES

☐ Enable non-standard use of single-DES in SSH 2

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

4. The window changes as shown in Figure 3-4 on page 20.
   a. Select the **Use background colour to erase screen** check box, which results in a better job of painting the screen for applications that uses curses (block graphics).

b. Click **Window** in the left pane.



Options controlling the terminal emulation

Set various terminal options

☑ Auto wrap mode initially on
☐ DEC Origin Mode initially on
☐ Implicit CR in every LF
☑ Use background colour to erase screen
☐ Enable blinking text

Answerback to ^E: PuTTY

Line discipline options

Local echo: ● Auto ○ Force on ○ Force off

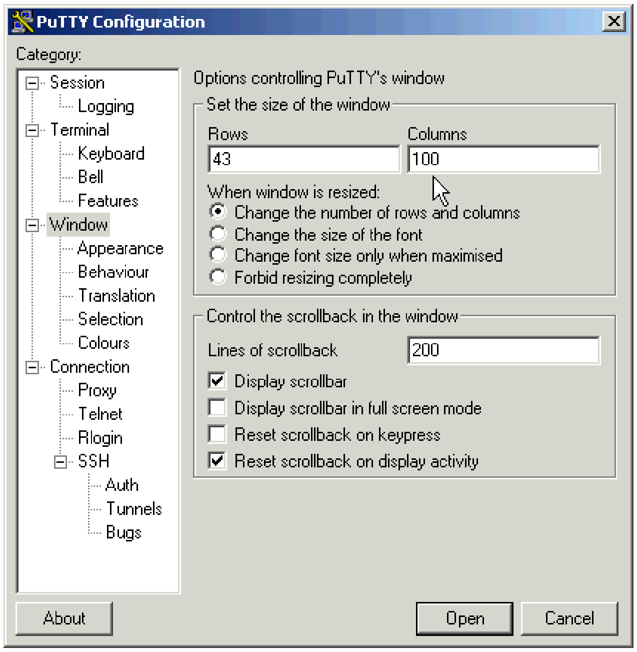
Local line editing: ● Auto ○ Force on ○ Force off

Remote-controlled printing

Printer to send ANSI printer output to: None (printing disabled)

Figure 3-4 Customizing PuTTY SSH settings (Part 2 of 4)

5. You see the window in Figure 3-5. You may choose more rows and columns than the default of 24x80. In this example, 43 rows and 100 columns are set. In the left panel, click **Session**.



Options controlling PuTTY's window

Set the size of the window

Rows: 43
Columns: 100

When window is resized:
● Change the number of rows and columns
○ Change the size of the font
○ Change font size only when maximised
○ Forbid resizing completely

Control the scrollback in the window

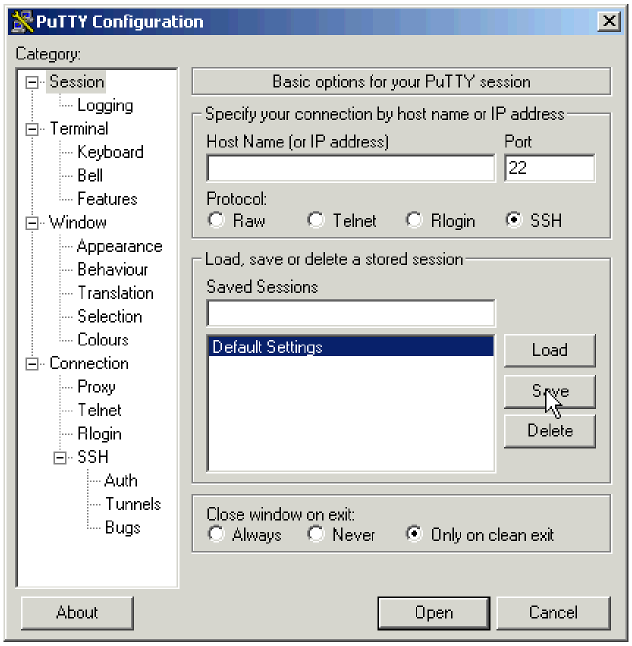
Lines of scrollback: 200

☑ Display scrollbar
☐ Display scrollbar in full screen mode
☐ Reset scrollback on keypress
☑ Reset scrollback on display activity

Figure 3-5 Customizing PuTTY window settings (Part 3 of 4)

6. You see the initial window when PuTTY is invoked (see Figure 3-6 on page 21).

   a. In the Save Sessions area, select **Default Settings** and click **Save**. This makes all future sessions that you define inherit these preferences.



Host Name (or IP address):

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

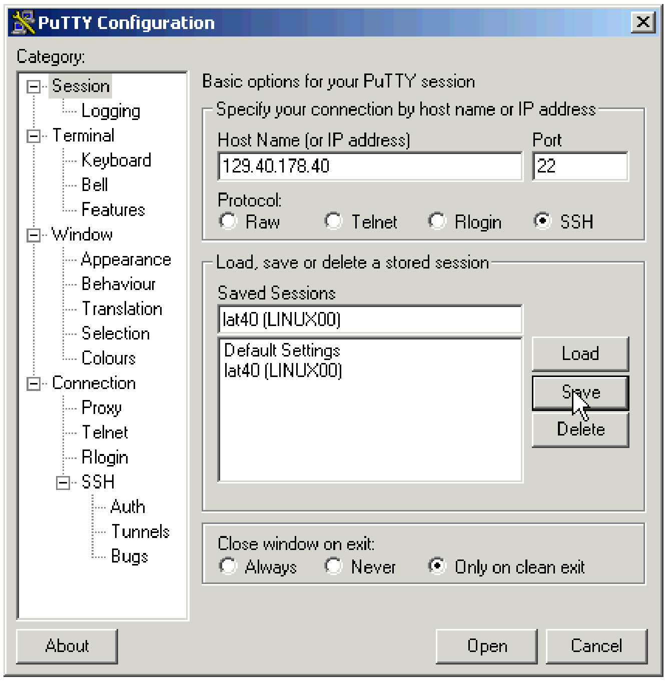
Figure 3-6 Customizing PuTTY window settings (Part 4 of 4)

b. In the Host Name (or IP address) field, enter the TCP/IP address (or DNS name).

c. Under Saved Sessions, choose a name that you will remember. In this example, the name `lat40 (SLES9)` is used. This is the DNS name and the z/VM user ID.

d. Again click **Save** and the session is saved.

Now whenever you start PuTTY, you can simply double-click any saved session name, and an SSH session for the desired Linux system. The final window looks similar to Figure 3-7.



Host Name (or IP address): 129.40.178.40

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions: lat40 (LINUX00)

Default Settings
lat40 (LINUX00)

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 3-7 Saving default PuTTY settings and creating new sessions

3.2 Setting up a VNC client

A VNC client allows access to a graphical environment with zSeries 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 recommended. You can purchase a full function client, or there is a free version. The RealVNC home page is:

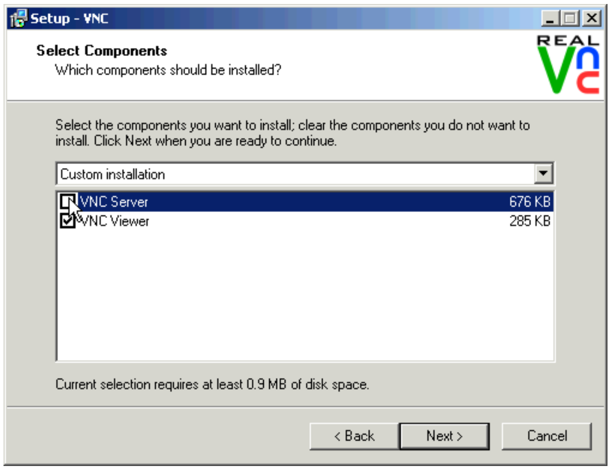
`http://www.realvnc.com/`

The page to download from 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 starts. At the time of writing of this book, RealVNC 4.1.1 is 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 panel as shown in Figure 3-8.



Setup - VNC

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom installation

☐	VNC Server	676 KB
☑	VNC Viewer	285 KB

Current selection requires at least 0.9 MB of disk space.

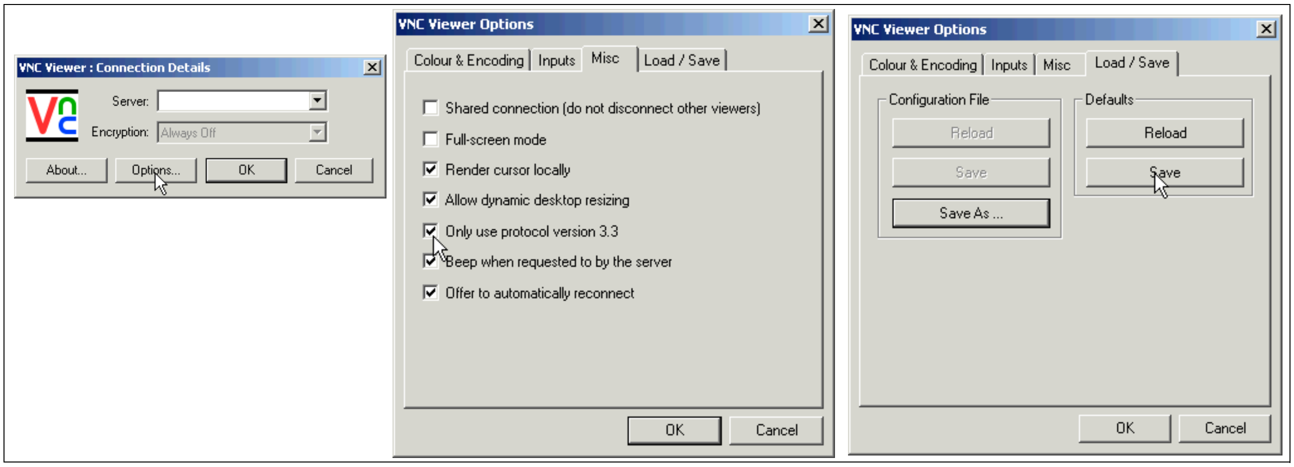
Figure 3-8 RealVNC Select Components panel

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. However, version 3.3 is still required with the VNC server shipped with SLES9. Therefore, to communicate, you must set the VNC protocol to 3.3. This should be the only configuration that is necessary.

Open the VNC client and click the **Options** button as shown in the left side of Figure 3-9 on page 23. Click the **Misc** tab. Click the check box named **Use only protocol version 3.3** as shown in the center of the figure. Finally, click the **Load/Save** tab and click **Save** the changes.



☐ Shared connection (do not disconnect other viewers)
☐ Full-screen mode
☑ Render cursor locally
☑ Allow dynamic desktop resizing
☑ Only use protocol version 3.3
☑ Beep when requested to by the server
☑ Offer to automatically reconnect

Figure 3-9 Setting VNC client to use protocol 3.3

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 Microsoft 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 characteristics of your emulator:

- Have the **Enter** and **Clear** keys 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 the Pause key in the upper right.
- Have a larger screen. Often the default number of lines in an emulator session is 24. You might prefer 32 or 43 lines, if that can fit easily in a window with your desktop display size and resolution.
- Set it so that the session automatically reconnects after logoff. Having a new logon screen 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 system to which you want to connect, spend a few minutes to define and save a session for each system to which you can connect, as was described for PuTTY. Then you can usually double-click the saved connection to quickly access a new 3270 session.

4

Installing and configuring z/VM

To complete this chapter, you must complete the majority of Chapter 6, “Configuring an NFS server” on page 75. However, it is recommended that you start here, because there is a step when installing z/VM (**`instdvd`**) that takes over two hours. While that process is running, you can configure the Network File System (NFS) server. Alternatively, if you have other colleagues who can work on the project, you can start both chapters at the same time on the different systems.

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 in 2.5.1, “z/VM resources worksheet” on page 14. You need access to the Hardware Management Console (HMC).

z/VM 5.2 is shipped on tape and DVD. z/VM should install faster from tape due to faster I/O speeds. Installing from tape might require more trips between the HMC and the tape drive.

If you are familiar with the HMC, you can use the two page *z/VM Summary for Automated Installation and Service (DVD Installation)* to replace or augment the description here.

If you are not familiar with the HMC and z/ VM, you might want to use the complete installation manual *z/VM Guide for Automated Installation and Service Version 5 Release 2.0*, GC24-6099. If you are installing z/VM at the second level (z/VM under z/VM) or onto a SCSI disk, use that book 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 5.2 from an HMC with a DVD-ROM onto 3390-3 DASD. For alternative configurations such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

1. On the Hardware Management Console, select the LPAR on which you want to install z/VM.
2. On the CPC Recovery menu, double-click the **Integrated 3270 Console** as shown at the bottom of Figure 4-1. A Personal Communications emulator session opens.

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

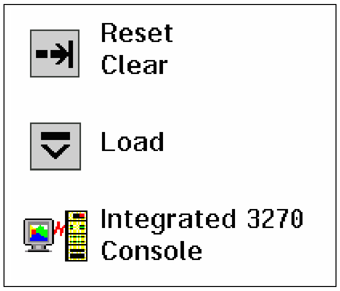


Figure 4-1 Integrated 3270 Console icon

3. Place the z/VM DVD Product Package 3390 in the HMC DVD drive.
4. Switch to Single Object Operations mode. To get into this mode, perform the following steps:
   a. Double-click **Defined CPCs** in the Groups Work Area.
   b. Select your CPC.
   c. If necessary, go around the racetrack (the buttons with circular arrows on the bottom right corner) to the CPC Recovery menu.

   d. Double-click the **Single Object Operations** icon. Click **yes** to confirm. Now the Primary Support Element Workplace™ window should appear. This is a window within a window.
   e. Double-click **Groups** near the top of this window.
   f. Double-click **Images** in the Groups Work Area.
5. Select the LPAR into which z/VM will be installed.
6. Go around the racetrack in this window to the CPC Recovery menu. Double-click the **Load from CD-ROM or Server** icon when you see it (see Figure 4-2).

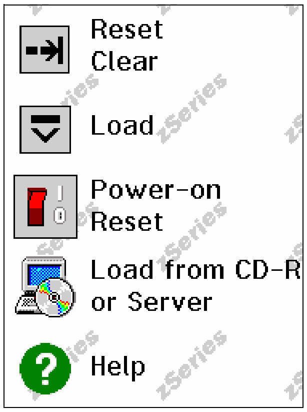
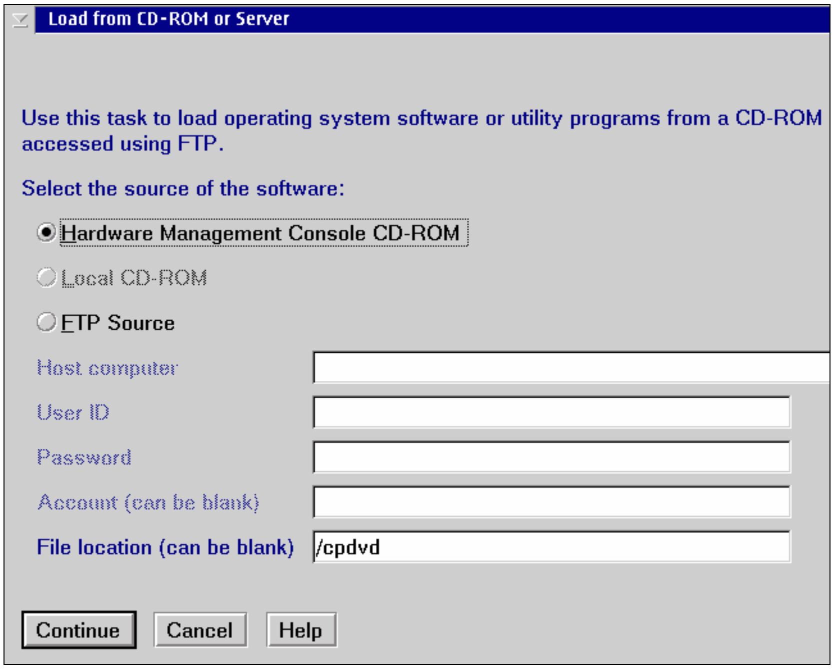


Figure 4-2 CPC Recovery menu with Load from CD-ROM or Server icon

7. Confirm you want to load by clicking **Yes**.
8. On the Load CD-ROM or Server window as shown in Figure 4-3 on page 28, the radio button **Hardware Management Console CD-ROM** should be selected.
9. In the same Load CD-ROM or Server window, fill in File Location with `/cpdvd`. This is the directory on the DVD with the z/VM 5.2 installation code.
10. Click **Continue**.


Load from CD-ROM or Server

Use this task to load operating system software or utility programs from a CD-ROM accessed using FTP.

Select the source of the software:

● Hardware Management Console CD-ROM

○ Local CD-ROM

○ FTP Source

Host computer

User ID

Password

Account (can be blank)

File location (can be blank) /cpdvd

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

11. Load the RAMDISK:
    a. From the *Load from CD-ROM or Server* panel, the software **520vm.ins** should be selected. Click **Continue**.
    b. From the Confirm the action window, click **Yes**. You should see the *Load from CD-ROM or Server Progress* window.
    c. When you see the message `Completed successfully`. Click **OK** to close. This should take about four to eight minutes.

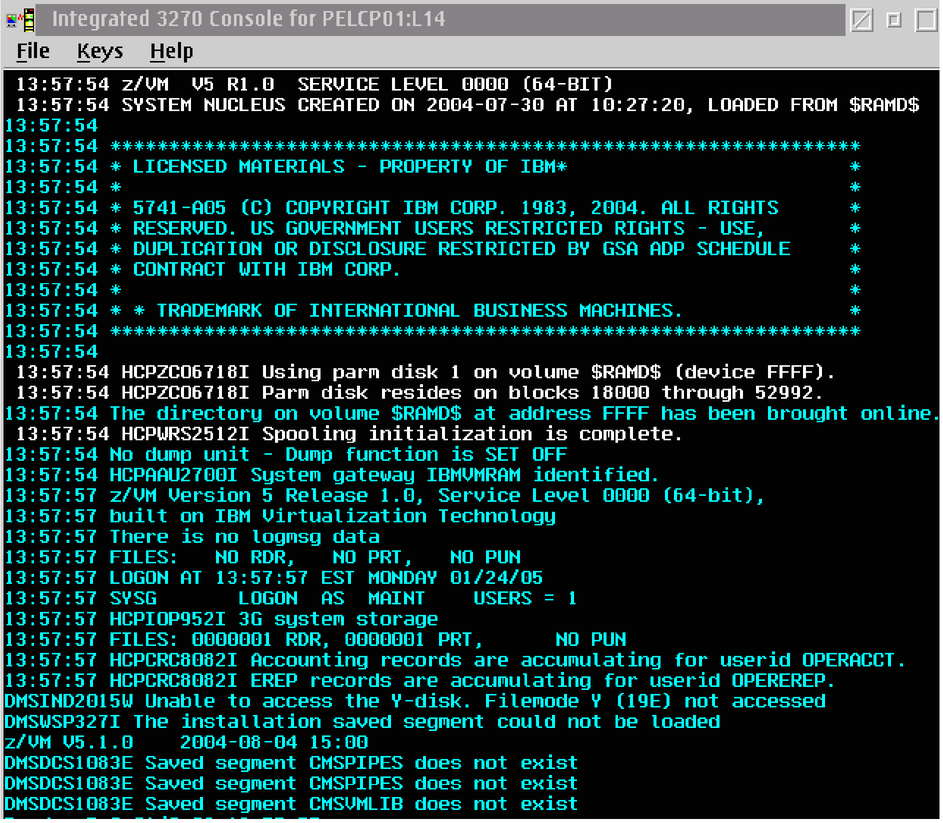
> **Important:** If you see the error `Unable to find software` and do not see the CD/DVD drive spin on the HMC, it is possible that your HMC firmware is down-level. This was a known issue with early HMC code and the IBM z9. Be sure your HMC service is up to the latest.

You should now have an in-memory z/VM 5.2 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. To do so, log off the primary SE window by closing the window within a window.
2. Use the **Alt-Tab** sequence, move to the Integrated 3270 Console window. The RAMdisk IPLs and the system comes up with the `MAINT` user ID logged on. You should see z/VM boot as shown in Figure 4-4:

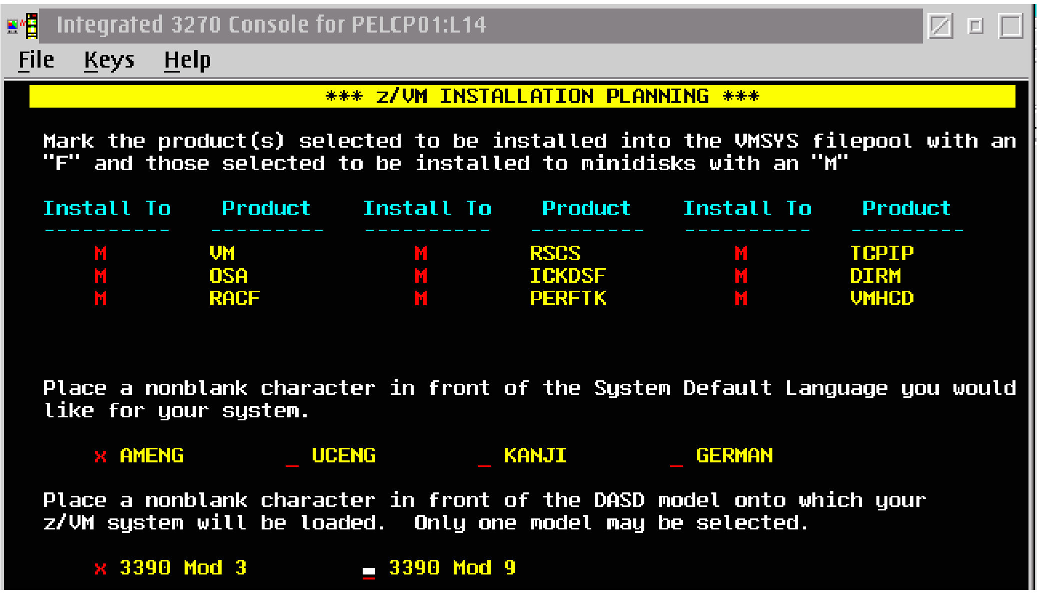


```
 13:57:54 z/VM  V5 R1.0  SERVICE LEVEL 0000 (64-BIT)
 13:57:54 SYSTEM NUCLEUS CREATED ON 2004-07-30 AT 10:27:20, LOADED FROM $RAMD$
13:57:54
13:57:54 ****************************************************************
13:57:54 * LICENSED MATERIALS - PROPERTY OF IBM*                        *
13:57:54 *                                                              *
13:57:54 * 5741-A05 (C) COPYRIGHT IBM CORP. 1983, 2004. ALL RIGHTS      *
13:57:54 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE,        *
13:57:54 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE     *
13:57:54 * CONTRACT WITH IBM CORP.                                      *
13:57:54 *                                                              *
13:57:54 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES.              *
13:57:54 ****************************************************************
13:57:54
 13:57:54 HCPZCO6718I Using parm disk 1 on volume $RAMD$ (device FFFF).
 13:57:54 HCPZCO6718I Parm disk resides on blocks 18000 through 52992.
13:57:54 The directory on volume $RAMD$ at address FFFF has been brought online.
 13:57:54 HCPWRS2512I Spooling initialization is complete.
13:57:54 No dump unit - Dump function is SET OFF
13:57:54 HCPAAU2700I System gateway IBMVMRAM identified.
13:57:57 z/VM Version 5 Release 1.0, Service Level 0000 (64-bit),
13:57:57 built on IBM Virtualization Technology
13:57:57 There is no logmsg data
13:57:57 FILES:   NO RDR,   NO PRT,   NO PUN
13:57:57 LOGON AT 13:57:57 EST MONDAY 01/24/05
13:57:57 SYSG       LOGON  AS  MAINT    USERS = 1
13:57:57 HCPIOP952I 3G system storage
13:57:57 FILES: 0000001 RDR, 0000001 PRT,      NO PUN
13:57:57 HCPCRC8082I Accounting records are accumulating for userid OPERACCT.
13:57:57 HCPCRC8082I EREP records are accumulating for userid OPEREREP.
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
DMSWSP327I The installation saved segment could not be loaded
z/VM V5.1.0    2004-08-04 15:00
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSVMLIB does not exist
```

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

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

   ```
   ==> instplan
   ```



```
                        *** z/VM INSTALLATION PLANNING ***

 Mark the product(s) selected to be installed into the VMSYS filepool with an
 "F" and those selected to be installed to minidisks with an "M"

 Install To    Product    Install To    Product    Install To    Product
 ----------   ---------   ----------   ---------   ----------   ---------
     M         VM              M        RSCS            M        TCPIP
     M         OSA             M        ICKDSF          M        DIRM
     M         RACF            M        PERFTK          M        VMHCD


 Place a nonblank character in front of the System Default Language you would
 like for your system.

     x AMENG          _ UCENG          _ KANJI          _ GERMAN

 Place a nonblank character in front of the DASD model onto which your
 z/VM system will be loaded.  Only one model may be selected.

     x 3390 Mod 3          _ 3390 Mod 9
```

Figure 4-5   Installation planning panel

4. You should see the display as shown in Figure 4-5 on page 29. During installation leave the M’s in the top section alone, and type the letter **X** next to `AMENG` (or select your language) and `3390 Mod 3` (or select 3390 Mod 9 if you are installing onto them) as shown above. Then press F5.

   If you choose to omit some products in the top section then blank out the M next to the products.

   You should the message `HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY`.

5. Attach the DASD devices onto which z/VM will be installed defined in your planning worksheet in 2.5.2, “z/VM DASD worksheet” on page 15. In this example, the devices are `a770-a774`.

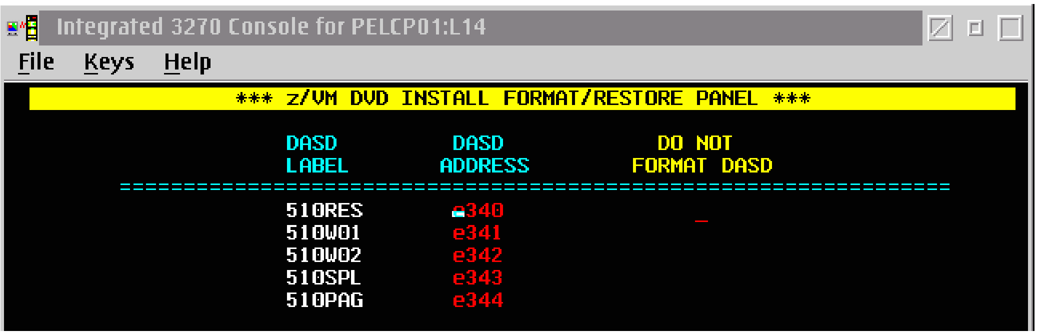
```
==> att <a770-a774> *
a770-a774 ATTACHED TO MAINT
```

> **Important:** The angle brackets, <> , in the above 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

6. Execute the **`INSTDVD EXEC`** to begin laying down z/VM to DASD:

```
==> instdvd
```

7. If you are using 3390-3s, you see a panel asking for the five volumes as shown in Figure 4-6.



```
               *** z/VM DVD INSTALL FORMAT/RESTORE PANEL ***

                  DASD          DASD            DO NOT
                  LABEL       ADDRESS         FORMAT DASD
         =================================================================
                  510RES        e340               _
                  510W01        e341
                  510W02        e342
                  510SPL        e343
                  510PAG        e344
```

Figure 4-6 INSTDVD DASD address panel

   a. Enter the addresses of the five volumes that onto which z/VM will be installed.
   b. Do *not* select the DO NOT FORMAT DASD check box on the right side of the panel.
   c. Press F5 to start the installation.

8. When you see the question `DO YOU WANT TO CONTINUE?`, type **Y.** You should see the message `NOW FORMATTING DASD <A770>`.

> **Important:** INSTDVD takes about two and one half hours. Now is be a good time to go to Chapter 6, “Configuring an NFS server” on page 75.

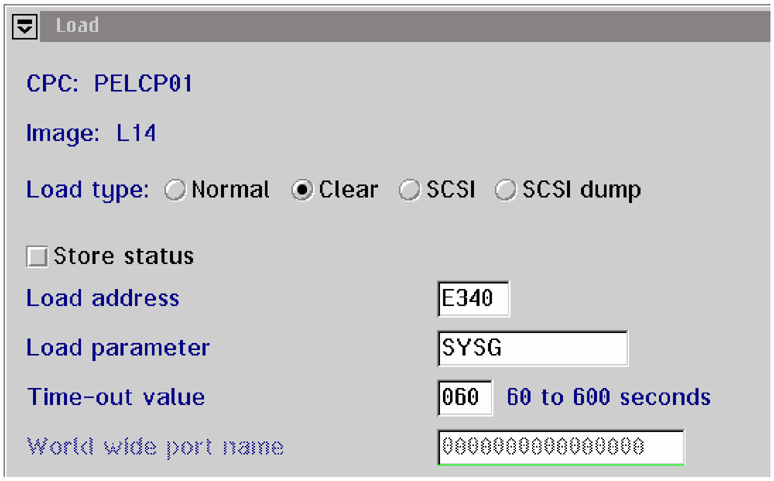
9. You are asked to place the system RSU in the drive. Insert it into the HMC DVD-ROM drive and type `GO`. You should see a messages of the form `DVDLOAD: LOADING FILE CKD500X IMAGE *.` This step takes two to four minutes.

10. 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.

1. From the HMC, **select your LPAR**. You may have to first double-click **Groups**.
2. You should see the CPC Recovery Menu. Double-click the **Load** icon in the menu at the right side.
3. The Load window opens as shown in Figure 4-7. Follow these steps:
   a. Check the radio button **Clear**.
   b. Set the load address to the new system residence (520RES) volume which is `<E340>` in this example.
   c. Set the load parameter to `SYSG`.
   d. Click **OK** to IPL.



CPC: PELCP01

Image: L14

Load type: ○ Normal ● Clear ○ SCSI ○ SCSI dump

☐ Store status

Load address: E340

Load parameter: SYSG

Time-out value: 060 60 to 600 seconds

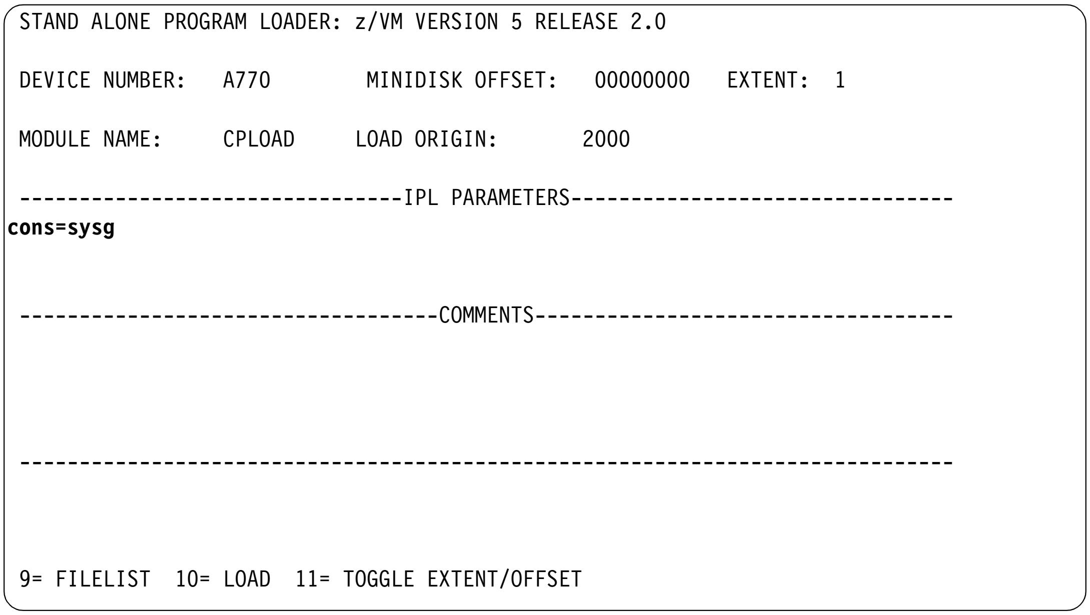
World wide port name: 0000000000000000

Figure 4-7 Load window

4. When you see the Load Task Confirmation window, click **Yes**.
5. After one to three minutes, you should see COMPLETED in the Load Program window. Click **OK**.
6. Use the **Alt-Tab** sequence to move back to the Integrated 3270 window. You should see the Standalone Program Loader panel as shown in Figure 4-12 on page 57.
   a. Press the **Tab** key to traverse to the IPL Parameters section and enter the value **`cons=sysg`**.

b. Press the F10 key to continue the IPL of your z/VM system. This should take around two to three minutes.

Figure 4-8 STAND ALONE PROGRAM LOADER



```
 STAND ALONE PROGRAM LOADER: z/VM VERSION 5 RELEASE 2.0

 DEVICE NUMBER:   A770       MINIDISK OFFSET:   00000000   EXTENT:  1

 MODULE NAME:     CPLOAD     LOAD ORIGIN:       2000

 --------------------------------IPL PARAMETERS--------------------------------
cons=sysg


 -----------------------------------COMMENTS-----------------------------------




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



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

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

```
==> 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`. You can disconnect from the `OPERATOR` user ID:

```
==> disc
```

Press Enter to get a new logon screen. This might take a minute or two.

4.1.4 Completing the z/VM installation

Follow these steps to complete the z/VM installation

1. On the z/VM login screen, logon as **`MAINT`.** The password is **`MAINT`**. You may receive messages `HCPLMN108E` or `DMSACP113S` 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.
2. IPL CMS and press Enter. Then run the **`instvm dvd`** command:

```
==> ipl cms:
==> Enter
==> instvm dvd
...
HCPPLD8329I POSTLOAD EXEC ENDED SUCCESSFULLY
...
```

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

3. Load the recommended service. For z/VM 5.2, the service name is 5202RSU1. Run the following commands:

```
==> ipl cms
==> Enter
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * servlink c
5201RSU1 SERVLINK C1
==> service all 5201rsu1
```

This step should take about 4-8 minutes. The last message should be `VMFSRV2760I SERVICE processing completed successfully`.

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

```
==> ipl cms
==> Enter
==> put2prod
```

This step should take about 4-8 minutes. The last message should be `VMFP2P2760I PUT2PROD processing completed successfully`.

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

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

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. Enter the following command to shutdown and reIPL your system:

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

6. You will lose your 3270 session. The system should come back in about two to four minutes. 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 screen.

Congratulations! You should now have a vanilla z/VM system installed.

4.2 Customizing the SYSTEM CONFIG file

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

- Change the system name
- Increase retrieve capacity and 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)

When your system comes back you should get a z/VM logon panel

1. **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 screen. After entering the user ID and password, press **Enter** when the status area in the lower right reads “`VM READ`**”.**

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

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

Example 4-1 QUERY CPDISK command

```
==> q cpdisk
```

```
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
==> 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    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
```

3. Once 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 to the minidisk.

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

4. Now the `MAINT CF1` disk is accessed read/write as your F disk. First make a backup copy of the vanilla `SYSTEM CONFIG` file using the **`COPYFILE`** command with the **`OLDDATE`** parameter so the file's time stamp is not modified, then edit the original copy:

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

5. The system name is set to `ZVMV5R10` by default in the `System_Identifier_Default` statement. You can search for it using the **`/ XEDIT`** subcommand:

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

Modify this to the new name of your system:

```
System_Identifier_Default <LNXVM52>
```

6. Next look for the `Features` statement. You can search for it again or you can use **`F8`** to page down. The following additions and changes 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:

Example 4-2

```
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  inifinite
```

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. If you have multiple z/VM systems, each should have a unique prefix. Modify the two starting addresses of the OSA triplets (`3004` and `3008` in this example) to those you specified in 2.5.1, “z/VM resources worksheet” on page 14.

```
====> bot
====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev <3004> <3008>
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 get 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:

Example 4-3 RELEASE with DETACH parameter and CPACCESS

```
==> 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.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
```

Note that all three CP disks are now accessed.

4.3 Configure the XEDIT profile

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 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.

One 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. It is recommended 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 adding the ? command:

```
SET PF12 FILE
```

After adding the **?** command :

```
SET PF12 ?
```

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

4.4 Configure TCP/IP

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

4.4.1 Use the IPWIZARD tool

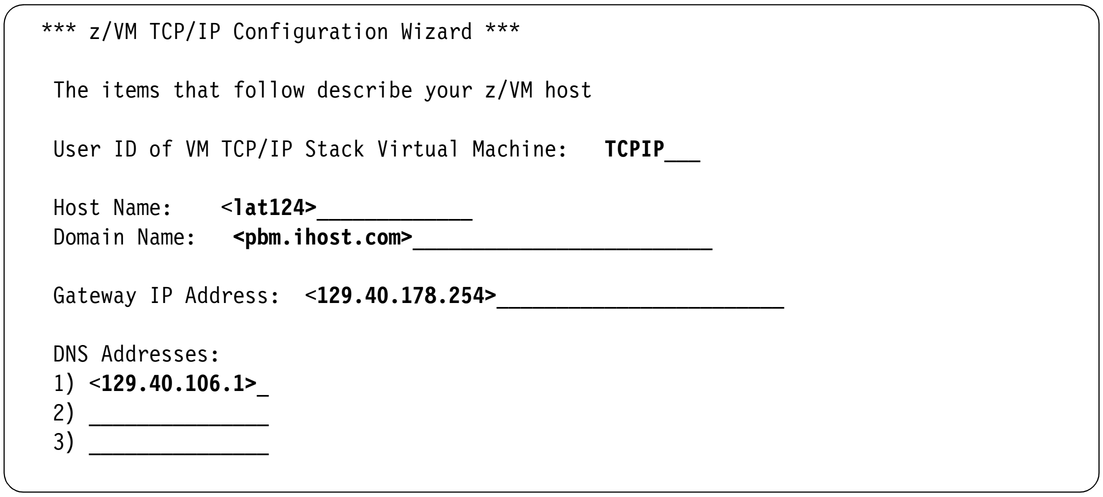
The **IPWIZARD** command is on the `MAINT 193` disk. You should already have this disk accessed as file mode G so you will pick up **IPWIZARD** from that minidisk.

1. Invoke **IPWIZARD**.

   ```
   ==> ipwizard
   ```

2. The *z/VM TCP/IP Configuration Wizard* opens as shown inFigure 4-9. The first field, User ID, should always be `TCPIP`. Obtain the remaining values from the 2.5.1, “z/VM resources worksheet” on page 14 and press F8.

Figure 4-9 TCP/IP Configuarion Wizard

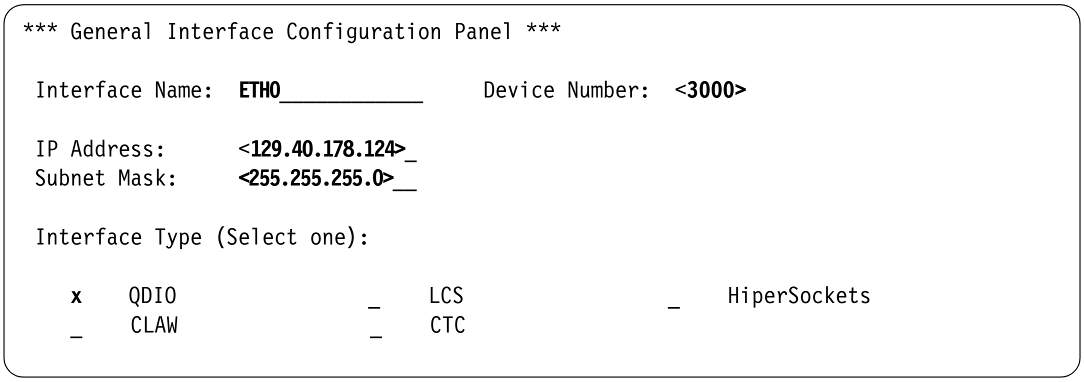


```
*** z/VM TCP/IP Configuration Wizard ***

 The items that follow describe your z/VM host

 User ID of VM TCP/IP Stack Virtual Machine:   TCPIP___

 Host Name:    <lat124>____________
 Domain Name:   <pbm.ihost.com>_______________________

 Gateway IP Address:  <129.40.178.254>_______________________

 DNS Addresses:
 1) <129.40.106.1>_
 2) ______________
 3) ______________
```

3. In Figure 4-10 on page 37, 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 which 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` with modern OSA devices. When completed, press F8.

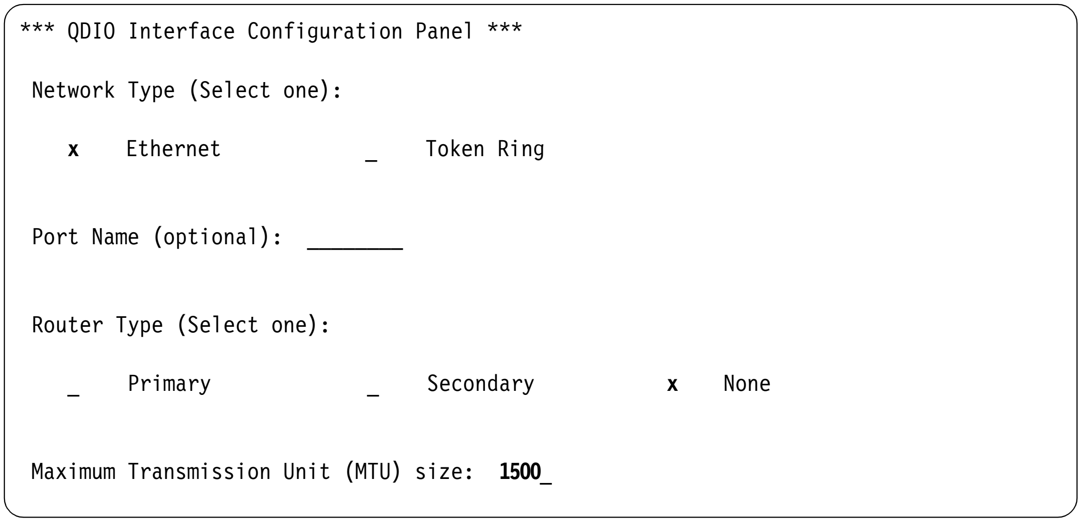
Figure 4-10 General Interface configuration



```
*** General Interface Configuration Panel ***

 Interface Name:  ETH0____________     Device Number:  <3000>

 IP Address:      <129.40.178.124>_
 Subnet Mask:     <255.255.255.0>__

 Interface Type (Select one):

    x    QDIO                 _    LCS                  _    HiperSockets
    _    CLAW                 _    CTC
```

4. 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.

Figure 4-11 QDIO Interface configuration



```
*** QDIO Interface Configuration Panel ***

 Network Type (Select one):

    x    Ethernet             _    Token Ring


 Port Name (optional):  ________


 Router Type (Select one):

    _    Primary              _    Secondary            x    None


 Maximum Transmission Unit (MTU) size:  1500_
```

You should see output similar to Example 4-4.

Example 4-4 Output of TCP/IP configuration wizard

```
DTCIPW2508I DTCIPWIZ EXEC is attempting to create the necessary
DTCIPW2508I configuration files
USER DSC   LOGOFF AS  TCPIP    USERS = 2     FORCED BY MAINT
...
Successfully PINGed Interface (129.40.178.124)
Successfully PINGed Gateway (129.40.178.254)
Ping Level 520: Pinging host 129.40.106.1.
                Enter 'HX' followed by 'BEGIN' to interrupt.
```

5. If the DNS server cannot be pinged, enter **`1`** to try it again. Watch for the message **`IPWIZARD EXEC ENDED SUCCESSFULLY`** (Example 4-5).

Example 4-5 Repinging the DNS server

```
PING: Ping #1 timed out
Not all of the PINGs were successful. Would you like
to try them again?
Enter 0 (No), 1 (Yes)
```

```
==> 1
...
Successfully PINGed Interface (129.40.178.124)
Successfully PINGed Gateway (129.40.178.254)
Successfully PINGed DNS (129.40.106.1)
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
```

6. At this point, your z/VM TCP/IP stack should be running. You should now 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.

4.4.2 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 `AUTOLOG1`'s **PROFILE EXEC**. If the **noautolog** parameter is not specified When z/VM starts the `AUTOLOG1` virtual machine is started. Because it 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 of `MAINT`.

   ```
   ==> log
   ```

2. You should see a new logon panel. 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. (If it does run, you will be logged off).

Example 4-6 access (noprof command

```
LOGON AUTOLOG1
z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:30:12 EST THURSDAY 01/19/06
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V5.2.0    2005-12-22 09:36
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.
   a. The z/VM Shared File System (SFS), is not required to run Linux so you can safely delete the three lines that **`XAUTOLOG`** the user IDs `VMSERVS`, `VMSERVR` and `VMSERVU`.
   b. You can also safely delete the `Address Command` line.
   c. Add a line to start the `TCPIP` user ID with the **`XAUTOLOG`** command and keep two statements that start the VSWITCH controllers.
   d. Add a line to **`logoff`** of `AUTOLOG1` when the PROFILE is complete, as in Example 4-7. There is no need to keep that virtual machine running as its sole purpose is to run the `PROFILE EXEC`.

Example 4-7 Changed AUTOLOG1

Before the added line:

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

Address Command
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
```

After the added line:

```
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'            /* start up TCPIP */
'CP XAUTOLOG DTCVSW1'          /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'          /* start VSWITCH controller 2 */
'cp logoff'                    /* logoff when done */
```

7. Save your changes with the **`FILE`** subcommand and **logoff of `AUTOLOG1`**:

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

4.4.3 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 chance that the `PROFILE TCPIP` file will be overwritten when applying maintenance.

Logon to `TCPMAINT`. The `PROFILE TCPIP` file is on the `TCPMAINT 198` disk which is accessed as the D disk.

Make a backup copy the original `PROFILE TCPIP`, then rename it to `<SYSTEM_ID> TCPIP` (where `<SYSTEM_ID>` is `LNXVM52` in this example). When the TCPIP service machine starts, it will search for this file before the file `PROFILE TCPIP`.

```
==> copy profile tcpip d = tcpiorig = (oldd
==> rename profile tcpip d <lnxvm52> = =
```

You have now renamed you TCP/IP profile.

4.4.4 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.

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
```

Now, XEDIT sessions on `TCPMAINT` will have a similar look and feel.

4.4.5 Configuring the FTP server

It is recommend that you turn on the FTP server. To do so, edit the newly renamed configuration file and add an `AUTOLOG` statement near the top of the file with `FTPSERVE` as the only entry. 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 in Example 4-8.

Example 4-8 Adding an AUOLOG statement to the renamed configuration file

```
==> x <lnxvm52> tcpip d
```

Before the change:

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

After the change:

```
; ----------------------------------------------------------------------
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
...
====> file
```

Save your changes with the `FILE` subcommand. You could continue to configure the system, but at this time it is recommended that you test your changes by shutting down and reIPLing the system.

4.4.6 Shutting down and reIPLing the system

It is now time to shutdown and reIPL the system. You should still be working from the HMC. Eventually you will want to work remotely over the network via a 3270 emulator. You will also want to be able to shutdown 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 Integrated Console session, but it should come back in a few minutes as described previously. You might want to watch the system console as z/VM shuts down and reinitializes.

   > **HMC Integrated 3270 Console or 3270 emulator?** At this point you can continue working at the HMC, or you can access your new system with a 3270 emulator over the network. See 3.2, “3270 emulators” on page 17 for a brief exposition of that subject. If you use a 3270 emulator, **`LOGON`** as `MAINT` now.

2. Login as `MAINT`. You should have TCP/IP and FTP access to z/VM.
3. Query the new VSWITCH (Example 4-9):

Example 4-9 Querying the new VSWITCH

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    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
```

You should see that the VSWITCH exists and that there are two built-in VSWITCH controllers, `DTCVSW1` and `DTCVSW2`. Before z/VM 5.2, these user IDs had to be created manually.

4. Query the changes made to the `Features` statement in the `SYSTEM CONFIG` file:

```
==> q vdisk userlim
VDISK USER   LIMIT IS INFINITE
```

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

4.5 Adding paging volumes

The z/VM operating system resides on the first three CP volumes. z/VM 5.2 now also installs with one full paging volume and one full spool volume. A single spool volume is probably adequate for Linux needs, however, a single paging volume probably is not. It is recommended that additional page volumes be created up front.

It is recommended that you add five paging volumes so you will have a total of six. If you do not have a lot of DASD, this number can be reduced. Having adequate paging space will give you plenty of space 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.

4.5.1 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 via 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** has been provided to allow you to format many volumes with a single command. The source code for this EXEC is in the section A.5.1, “The CPFORMAT EXEC” on page 211. It is a wrapper around **CPFMTXA**. To use this EXEC, each DASD to be formatted must first be attached with the virtual device address the same real 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.

Important: At this point, you need access to the server described in Chapter 5, “Configure an NFS server” on page 55 in order to get the files **CPFORMAT EXEC**. Be sure those steps are completed.

Getting the CPFORMAT EXEC to z/VM

Logoff of `MAINT` so you will be able to get the `MAINT 191` disk in read/write mode with FTP.

Start an SSH (PUTtY) session on the NFS server where the files associated with this book are were copied to (in 6.1, “Downloading files associated with this book” on page 76). Copy the `CPFORMAT.EXEC` from that server to z/VM with an FTP client (Example 4-10 on page 43). Being able to FTP to z/VM shows that both TCP/IP and the FTP server have started.

Example 4-10 Copying CPFORMAT.EXEC from NFS to z/VM

```
# cd /nfs/virt-cookbook/vm
# ftp <129.40.178.124>
220-FTPSERVE IBM VM Level 520 at LAT124.PBM.IHOST.COM, 14:53:44 EST WEDNESDAY 2004-12-08
Name (129.40.178.124:root): maint
Password:
...
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 **`?`**, as in Example 4-11.

Example 4-11 Accessing CPFORMAT help

```
==> cpformat ?

Synopsis:

  Format one or a range of DASD as page, perm, spool or temp disk space
  The label written to each DASD is V<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-------'       '-TDSK-'
```

Example 4-12 shows how to attach five DASD volumes and use **`CPFORMAT`** to format them as paging space (refer to the planning work sheets that you filled out in 2.5.2, "z/VM DASD worksheet" on page 15): 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:

Example 4-12 Five DASD volumes using CPFORMAT

```
==> att <a775-a779> *
A775-A779 ATTACHED TO MAINT
==> cpformat <a775-a779> as page

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A775 MAINT    A775 3390  VPA775 A775          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A776 MAINT    A776 3390  VPA776 A776          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A777 MAINT    A777 3390  VPA777 A777          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A784 MAINT    A784 3390  VPA784 A784          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A779 MAINT    A779 3390  VPA779 A779          0       3339

WARNING - this will destroy data!
```

```
ARE YOU SURE you want to format the DASD as PAGE space (y/n)?
y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A775 MAINT    A775 3390  VPA775 A775          0       3339
MAINT    A776 MAINT    A776 3390  VPA776 A776          0       3339
MAINT    A777 MAINT    A777 3390  VPA777 A777          0       3339
MAINT    A784 MAINT    A784 3390  VPA784 A784          0       3339
MAINT    A779 MAINT    A779 3390  VPA779 A779          0       3339
```

This formatting job should run for about 10-50 minutes depending on many factors. But do not take a break now! You can format more volumes for PERM (minidisk) space in the next section.

4.5.2 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** (Example 4-13).

Example 4-13 3270 session as SYSMAINT

```
LOGON SYSMAINT
z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 11:13:41 EST WEDNESDAY 01/18/06
z/VM V5.2.0    2005-12-22 09:36

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. Attach the seven volumes that you will use for the `SLES9X` user ID. In Example 4-14, it is the DASD at addresses A781-A787. Invoke the **`CPFORMAT`** command against these volumes using the parameter **`as perm`**:

Example 4-14 Attaching seven permanant DASD volumes

```
==> att <a781-a787> *
A781-A787 ATTACHED TO MAINT
==> cpformat <a781-a787> as perm

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A781 MAINT    A781 3390  VMA781 A781          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A782 MAINT    A782 3390  VMA782 A782          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A783 MAINT    A783 3390  VMA783 A783          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
```

```
MAINT    A784 MAINT    A784 3390  VMA784 A784          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A785 MAINT    A785 3390  VMA785 A785          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A786 MAINT    A786 3390  VMA786 A786          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A787 MAINT    A787 3390  VMA787 A787          0       3339

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PAGE space (y/n)?
y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A781 MAINT    A781 3390  VMA781 A781          0       3339
MAINT    A782 MAINT    A782 3390  VMA782 A782          0       3339
MAINT    A783 MAINT    A783 3390  VMA783 A783          0       3339
MAINT    A784 MAINT    A784 3390  VMA784 A784          0       3339
MAINT    A785 MAINT    A785 3390  VMA785 A785          0       3339
MAINT    A786 MAINT    A786 3390  VMA786 A786          0       3339
MAINT    A787 MAINT    A787 3390  VMA787 A787          0       3339
```

Now you can take a break! You should now have page volumes being formatted on `MAINT` and PERM or minidisk volumes being formatted on `SYSMAINT`.

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

4.5.3 Updating the SYSTEM CONFIG file

Follow these steps to update the `SYSTEM CONFIG` file:

1. Now that the many PAGE and PERM volumes are ready for use, they must be added to the `SYSTEM CONFIG` file so z/VM knows about them. Example 4-15 uses the same steps to access the `MAINT CF1 disk` read/write that you used earlier.

Example 4-15 Updating the SYSTEM CONFIG with the DASD

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
```

   It is good to remember this sequence of steps.

2. Edit the `SYSTEM CONFIG` file and specify each of the new page volumes (`PAGE`) by name as `CP_Owned` (Example 4-16). When you system IPLs it will pick up these as paging volumes.

Example 4-16 Editing SYSTEM CONFIG

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

```
/*****************************************************************/

    CP_Owned   Slot   1  520RES
    CP_Owned   Slot   2  520W01
    CP_Owned   Slot   3  520W02
    CP_Owned   Slot   4  520SPL
    CP_Owned   Slot   5  520PAG
    CP_Owned   Slot   6  <VPA775>
    CP_Owned   Slot   7  <VPA776>
    CP_Owned   Slot   8  <VPA777>
    CP_Owned   Slot   9  <VPA778>
    CP_Owned   Slot   10  <VPA779>
    CP_Owned   Slot   11  RESERVED
    CP_Owned   Slot   12  RESERVED
    CP_Owned   Slot   13  RESERVED
...
```

3. Move down to the `User_Volume_List` section (Example 4-17). User volumes (`PERM`) can be specified individually with the `User_Volume_List` statement, or with wild cards via 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 “`VM`”:

Example 4-17 Editing User_Volume_List

```
/*********************************************************************/
/*                         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 VM*
/*  User_Volume_List USRP01    */
/*  User_Volume_List USRP02    */
...
```

4. Save your changes with the **`FILE`** subcommand. Verify the integrity of the changes with the **`CPSYNTAX`** command and put the `MAINT CF1` disk back online (Example 4-18). The following example shows how to do this as you did earlier:

Example 4-18 Verifying changes

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> 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.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
```

4.5.4 Testing the changes

It is recommended that you again shutdown and reIPL to test the changes. Before you shut down, note that you have only one page volume (`520PAG`) through the **`QUERY ALLOC PAGE`** command. Your output should look similar to Example 4-19.

Example 4-19 Testing changes with QUERY ALLOC PAGE

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
520PAG A772          1       3338 600840     12     12   1%
                                  ------ ------        ----
SUMMARY                           601020     12          1%
USABLE                            601020     12          1%
```

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 via a 3270 emulator, you will lose your session, but if all goes well, your system will be available again in a couple of minutes.

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

After the system comes back, **logon as MAINT** and look at the page space again. You should see output similar to Example 4-20.

Example 4-20 QUERY ALLOC PAGE viewed as MAINT

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
520PAG A772          1       3338 600840      0      0   0%
VPA775 A775          0       3338 601020      0      0   0%
VPA776 A776          0       3338 601020      0      0   0%
VPA777 A777          0       3338 601020      0      0   0%
VPA778 A778          0       3338 601020      0      0   0%
VPA779 A779          0       3338 601020     12     12   1%
                                  ------ ------        ----
SUMMARY                            3521K     12          1%
USABLE                             3521K     12          1%
```

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

4.6 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 both CMS and Linux users. Before starting, make a copy of the original `USER DIRECT` file:

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

4.6.1 Define the user in the USER DIRECT file

A small 20-cylinder minidisk is allocated at virtual address 191 and a larger 300 cylinder minidisk, to be shared by many guests, is defined at virtual address 192. Use the next free

DASD designated as PERM space on your worksheet (2.5.2, “z/VM DASD worksheet” on page 15). Cylinder 0 should always be reserved for the label therefore you should start minidisks at cylinder 1.

1. Edit the `USER DIRECT` file and add the user ID definition in Example 4-21 to the bottom of the file:

Example 4-21 User ID definition

```
==> 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 <VMA783> MR READ WRITE MULTIPLE  4
 MDISK 0192 3390 0021 0300 <VMA783> MR ALL WRITE MULTIPLE   5
*                                                           6
...
====> file
```

Note the following points for the numbers in black:

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

2 Include the profile named `TCPCMSU`

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 `VMA783`

5 Define 192 minidisk of `size 300` cylinders from volume `VMA783` 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 you must always check for overlapping cylinders and gaps (gaps will only leave empty disk space, however z/VM will allow you to *shoot yourself in the foot* by defining multiple minidisks over the same disk space). This is done with 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, as in Example 4-22 on page 49.

Example 4-22 Gaps between cylinders

```
====> /gaps
-------------------------------------------------------------------------

VOLUME   USERID      CUU   DEVTYPE      START        END         SIZE
                                            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
VMA783   LNXMAINT   0191     3390       00001      00020        00020
         LNXMAINT   0192     3390       00021      00320        00300
...
```

The two GAPs should be listed on the right side: a gap of 501 cylinders on the `$$$$$$` volume and a new gap of 1 cylinder exists on the volume that was just used to create disk space for the `LNXMAINT` user ID; in this case the `VMA783` volume.

Do not worry about the 501 cylinder gap, but to avoid a 1 cylinder gap being reported on each user volume, it is recommended to 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, rather, it is a convenient place to put dummy minidisk definitions for cylinder 0 of all `PERM` volumes.

Look at the rest of the file. You should see the three volumes that z/VM installs onto are already there (`520RES, 520W01, 520W02`).

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, as in Example 4-23.

Example 4-23 Adding a new minidisk definition to USER DIRECT

```
==> x user direct
====> /user $alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 520RES R
 MDISK A02 3390 000 001 520W01 R
 MDISK A03 3390 000 001 520W02 R
MDISK A04 3390 000 001 <VMA783> R
```

7. Save your changes and run **`DISKMAP`** again. Edit the `USER DISKMAP` file. This time you should see just the single 501 cylinder gap and cylinder 0 of the first user volume allocated to the `$ALLOC$` user ID. When you are done you can quit without saving changes by pressing F3.

Example 4-24 Editing USER DISKMAP

```
==> diskmap user
==> x user diskmap
====> prefix off
====> /$ALLOC
...
VOLUME   USERID      CUU   DEVTYPE      START        END         SIZE
VMA783   $ALLOC$     A04     3390       00000      00000        00001
         LNXMAINT   0191     3390       00001      00020        00020
         LNXMAINT   0192     3390       00021      00320        00300
...
====> F3
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 39 disk pages
```

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

You have now defined your first z/VM user ID named `LNXMAINT`.

4.6.2 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 of `MAINT`** and **logon to `LNXMAINT`**, as in Example 4-25.

Example 4-25 Log on the LNXMAINT

```
LOGON LNXMAINT
z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 12:52:24 EST WEDNESDAY 01/18/06
z/VM V5.2.0    2005-12-22 09:36

DMSACP112S A(191) device error
```

You should see an error message ending in “`device error`”. 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`** in Example 4-26.

Example 4-26 Formatting a disk for CMS

```
==> 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 (Example 4-27) which should take a minute or two:

Example 4-27 Formatting 192 disk as the D minidisk

```
==> 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)
```

You have now formatted the two minidisks and accessed them as file modes `A` and `D`.

4.6.3 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:

DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

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

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

4.6.4 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 and add the following lines in Example 4-28. REXX EXECs must always begin with a C language-style comment.

Example 4-28 Creating a PROFILE EXEC

```
==> 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` access the `TCPMAINT 592` disk as file mode E, sets CP run on, and sets the retrieve keys per the convention.

2. You could test your changes by logging off and logging back on. However, typing the command **`PROFILE`** will do the same. 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`. To see your minidisks, use the **`QUERY DISK`** command as in Example 4-29.

Example 4-29 Using QUERY DISK

```
==> profile
DMSACP723I E (592) R/O
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        2          9-01       3591       3600
LXM192 192  D   R/W   300 3390 4096       14       3747-02     176253     180000
TCM592 592  E   R/O    67 3390 4096      877       8167-68       3893      12060
MNT190 190  S   R/O   100 3390 4096      689      14325-80       3675      18000
MNT19E 19E  Y/S R/O   250 3390 4096     1010      26665-59      18335      45000
MNT191 191  Z   R/O   175 3390 4096       36        224-01      31276      31500
```

3. Verify that your F11 and F12 keys are set to the **`RETRIEVE`** command:

```
==> q pf11
PF11 RETRIEVE FORWARD
```

```
==> q pf12
PF12 RETRIEVE BACKWARD
```

4.6.5 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. **Log off of 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.

```
# cd /nfs/virt-cookbook/vm
```

FTP to z/VM. By default FTP copies files to your 191 disk, so first change directory to the `LNXMAINT 192` disk. Then use the **`mput *`** subcommand to copy all the files from the `vm/` subdirectory to `LNXMAINT`. The files are all in ASCII so the default transfer type of ASCII will cause the files to be converted to EBCDIC.

Example 4-30 Copying files to LNXMAINT 192

```
# ftp <129.40.178.124>
220-FTPSERVE IBM VM Level 520 at LAT124.PBM.IHOST.COM, 14:53:44 EST WEDNESDAY 2004-12-08
Name (129.40.178.124:root): lnxmaint
Password:
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> prompt
Interactive mode off
ftp> mput *
...
ftp> quit
```

Logon to `LNXMAINT`. You should see the files in Example 4-31 on your D disk.

Example 4-31 Files copied to LNXMAINT

```
==> 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
      CHPW52   XEDIT    D1 V         70        180          3  1/24/06 14:48:15
      CPFORMAT EXEC     D1 V         79        231          3  1/24/06 14:48:15
      LABEL520 EXEC     D1 V         73        112          2  1/24/06 14:48:15
      LABEL520 XEDIT    D1 V         71         19          1  1/24/06 14:48:15
      PROFILE  EXEC     D1 V         71         21          1  1/24/06 14:48:15
      SLES9X   EXEC     D1 V         74          9          1  1/24/06 14:48:15
      SLES9X   PARMFILE D1 V         62          8          1  1/24/06 14:48:15
      SWAPGEN  EXEC     D1 V         72        358          5  1/24/06 14:48:15
```

See Appendix A.5, “z/VM source code listings” on page 211 for more details.

4.7 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.

4.7.1 Configuring the AUTOLOG1 PROFILE EXEC

It is recommended that the following tasks be accomplished by using `AUTOLOG1`'s **`PROFILE EXEC`**.

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

1. **Logoff of `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`**, as in Example 4-32.

Example 4-32 Running ACCESS (NOPROF on AUTOLOG1

```
LOGON AUTOLOG1
z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:39:10 EST WEDNESDAY 01/18/06
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V5.2.0    2005-12-22 09:36
==> acc (noprof
```

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

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

3. Edit the file and add the emboldened text in Example 4-33.

Example 4-33 Editing PROFILE EXEC

```
==> x profile exec
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'               /* start up TCPIP */
'CP XAUTOLOG DTCVSW1'             /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'             /* start VSWITCH controller 2 */
'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 180'      /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant sles9x'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog sles9x'

'cp logoff'                        /* logoff when done */
```

4. 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 good starts for Linux systems, but may not be optimal. For more reading on 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`
`http://www.zjournal.com/PDF/robinson.pdf`

You can 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 the two `SLES9` user IDs.

4.7.2 Testing the changes

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

1. Shutdown and reIPL your system.

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

2. When your system comes back **logon as MAINT**.
3. Use the **`QUERY NAMES`** command to see that `TCPIP`, the FTP server and the two VSWITCH controllers have been logged on, as in Example 4-34.

Example 4-34 Using QUERY NAMES

```
==> q n
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
MAINT    -L0004
VSM     - TCPIP
```

4. Query the SRM values to see that the new `STORBUF` settings is in effect and the `SIGNAL SHUTDOWN` value is set to 180 seconds, as in Example 4-35.

Example 4-35 Using QUERY SRM

```
==> 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: 180 seconds
```

This output shows that your changes have taken effect.

4.8 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

VM security products

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

High level z/VM security

The paper *z/VM Security and Integrity* by Cliff Laking and Alan Altmark discusses the isolation and integrity of virtual servers under z/VM. It is 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 *Running Linux Guests with less than CP Class G Privilege* by Rob van der Heij 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 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 `CHPW52 XEDIT` macro defined in the next section

4.8.1 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 `CHPW52 XEDIT` has been included with this book. The source code is in Appendix A.5.2, "The CHPW52 XEDIT macro" on page 215.

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 CHPW52 EXEC.

```
==> 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 good candidate for a password.

4. Edit the `USER DIRECT` file with a parameter of **`(profile chpw52)`** followed by the new password. Rather than invoking the default profile of `PROFILE XEDIT`, this command will invoke the **`XEDIT`** macro named `CHPW52 XEDIT` and pass it the new password. For example, to change all passwords to `lnx4vm`, enter the following command:

```
==> x user direct c (profile chpw52) 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. 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 2.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 39 disk pages
```

Your new directory is online. Do not forget 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 as you can use the **`XEDIT CHANGE`** subcommand. For example to change all passwords from `lnx4vm` to `vm5lnx`, invoke the following commands:

```
==> x user direct c
====> c/lnx4vm/vm5lnx/* *
DMSXCG517I 773 occurrence(s) changed on 328 line(s)
```

Congratulations, your z/VM system is now customized and ready for Linux. It is recommended that you back up your system to tape.

4.9 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 with a user ID for shared files. You have changed the passwords. This would be a good time to back up the system to tape.

There are five system volumes that should be backed up 520RES, 520W01, 520W02, 520SPL and 520PAG. 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 (`VMA783` in this example). Back up that entire volume, because the remainder of it will be used for Linux backup data also.

To backup these volumes to tape, refer to chapter 8. *Load the System Image*, Step 11. *Store a Backup Copy of the z/VM System on Tape* in the manual *The z/VM Guide for Automated Installation and Service*, GC204-6099.

4.10 Relabel the system volumes

This step is optional, however, it is recommended. There are times when you will want to change the volume labels of the five z/VM system volumes. If there is a possibility that another z/VM system with the same labels is installed onto volumes accessible by your z/VM system, one of the systems will not IPL correctly.

To understand this possibility, refer to Figure 4-12. The z/VM system with the lower device addresses starting at E340 should IPL fine (though 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 520RES volume will be used, but the remaining volumes in the system are searched for by volume label, not by device address. Therefore, z/VM system2 will be using z/VM system1's volumes. This is not good.

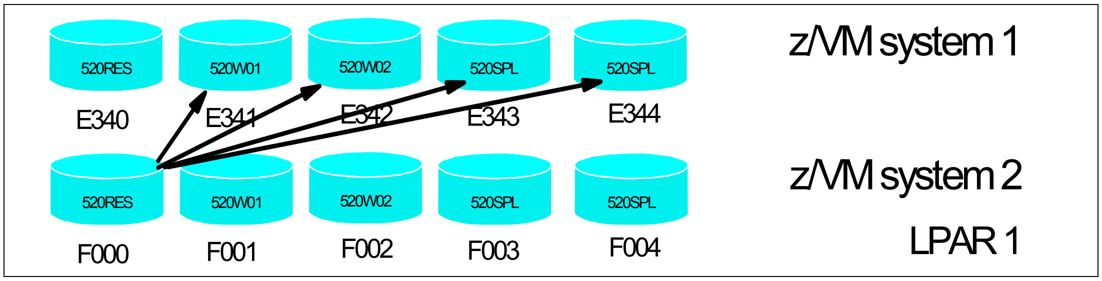


Figure 4-12 The problem with two z/VM systems with identical volume labels

So 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:

- 4.10.1, "Modifying labels in the SYSTEM CONFIG file" on page 57
- 4.10.2, "Modifying labels in the USER DIRECT file" on page 59
- 4.10.3, "Changing the labels on the five volumes" on page 59
- 4.10.4, "Shutting down your system and restarting it" on page 60

Important: This process must be done as documented. Making a mistake in one of the steps can easily result in an unusable system. Check your steps carefully and your system will come back with no problems. Try to do all steps in succession in a short amount of time.

4.10.1 Modifying labels in the SYSTEM CONFIG file

Note the first five CP-owned volumes with the QUERY CPOWNED command in Example 4-36.

Example 4-36 QUERY CPWONED output

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  520RES  A770  Own    Online and attached
   2  520SPL  A771  Own    Online and attached
   3  520PAG  A772  Own    Online and attached
   4  520W01  A773  Own    Online and attached
   5  520W02  A774  Own    Online and attached
   6  VPA775  A775  Own    Online and attached
   7  VPA776  A776  Own    Online and attached
   8  VPA777  A777  Own    Online and attached
   9  VPA778  A778  Own    Online and attached
   10  VPA779  A779  Own    Online and attached
   11  ------  ----  -----  Reserved
   12  ------  ----  -----  Reserved
   ...
```

The labeling convention described in 2.2.1, "Volume labeling convention" on page 9 suggests using 'V' in the second character of the label. An **XEDIT** macro, `LABEL520 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, then edit it with the **LABEL520 XEDIT** macro passing the five addresses of the z/VM system volumes (`a770-a774` in Example 4-37):

Example 4-37 Editing the labels in SYSTEM CONFIG

```
==> cprel a
...
==> link * cf1 cf1 mr
==> acc cf1 f
==> copy system config f = confwrks = (oldd rep
==> x system config f (profile label520) <a770> <a771> <a772> <a773> <a774>
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)
```

Clear the screen and you will be left editing the file. Search for the string `cp_owned` and you should see the new labels (Example 4-38). Be sure they are correct before saving the file with the **FILE** subcommand:

Example 4-38 Renamed files

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

    CP_Owned   Slot   1  VVA770
    CP_Owned   Slot   2  VVA773
    CP_Owned   Slot   3  VVA774
    CP_Owned   Slot   4  VVA771
    CP_Owned   Slot   5  VVA772
    CP_Owned   Slot   6  VPA776
    CP_Owned   Slot   7  VPA778
    CP_Owned   Slot   8  VPA775
    CP_Owned   Slot   9  VPA777
====> file
```

Verify there are no syntax errors:

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

Release and detach the `F` disk, **CPACCESS** the A disk and verify, as in Example 4-39.

Example 4-39 Releasing and detaching the F disk, and verifying the A disk

```
==> rel f (det
DASD 0CF1 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 0CF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES A770 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES A770 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES A770 CKD         129        188
```

You have now changed the labels of the system volumes in the `SYSTEM CONFIG` file. It is critical that you proceed as your system is now in a state where it will not reIPL.

4.10.2 Modifying labels in the USER DIRECT file

Now modify the labels in the `USER DIRECT` file. You should see many more occurrences of the labels being changed, as in Example 4-40.

Example 4-40 Modifying labels in USER DIRECT

```
==> copy user direct c = direwrks = (oldd rep
==> x user direct c (profile label520) <a770> <a771> <a772> <a773> <a774>
DMSXCG517I 84 occurrence(s) changed on 84 line(s)
DMSXCG517I 134 occurrence(s) changed on 134 line(s)
DMSXCG517I 69 occurrence(s) changed on 69 line(s)
DMSXCG517I 2 occurrence(s) changed on 2 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

You can choose to traverse the file before saving the changes:

```
====> 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.

4.10.3 Changing the labels on the five volumes

Next change the labels on the five volumes using the **CPFMTXA** command. You could do this one volume at a time with the **CPFMTXA LABEL** command. However, the **LABEL520** EXEC (Example 4-41) has been written to make this step easier. It takes the physical addresses of the five system volumes and applies the labelling convention used in this book.

Example 4-41 LABEL520 EXEC

```
==> label520 <a770> <a771> <a772> <a773> <a774>
The volumes are:
DASD A770 CP OWNED  520RES   49
DASD A771 CP OWNED  520W01   89
DASD A772 CP OWNED  520W02   1
DASD A773 CP OWNED  520SPL   1
DASD A774 CP OWNED  520PAG   0

The system volume labels will become:
VVA770 VVA771 VVA772 VVA773 VVA774

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

```
...
ICK03000I CPVOL REPORT FOR 0123 FOLLOWS:

          VOLUME SERIAL NUMBER IS NOW = VVA770
...
VOLUME SERIAL NUMBER IS NOW = VVA771
...
VOLUME SERIAL NUMBER IS NOW = VVA772
...
VOLUME SERIAL NUMBER IS NOW = VVA773
...
VOLUME SERIAL NUMBER IS NOW = VVA774
...
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
DASD 0A03 DETACHED
```

Now that the five volumes have been relabeled (sometimes called *clipping the volumes*), you can run the **DIRECTXA** command to update the directory:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
EOJ DIRECTORY UPDATED
HCPDIR494I User directory occupies 39 disk pages
Ready(00005); T=0.01/0.01 14:30:37
```

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.

4.10.4 Shutting down your system and restarting it

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
```

You will lose your 3270 session. Perform the following steps to bring the system back up:

1. Go back to the HMC to IPL your system.
2. Click the **LOAD** icon in the *CPC Recovery* menu.
3. Select the **Clear** radio button. All the other parameters should be correct from the previous IPL. Click **OK**
4. Click **Yes** on the *Load Task Confirmation* panel.
5. Go back to the Integrated 3270 console. After a few minutes the *Standalone Program Loader* panel should appear. Use the **TAB** key to traverse to the section *IPL Parameters* and enter the value **cons=sysg**
6. Press the F10 key to continue the IPL of your z/VM system. This takes around three minutes.
7. At the Start prompt you have to specify a FORCE start, again because the spool volume label has changed. Enter the following:

   ```
   ==> force drain
   ```
8. Do not change the time of day clock.

```
==> no
```

9. When the IPL completes, **`DISCONNECT`** from the `OPERATOR` user ID and **logon to** `MAINT`.

```
==> disc
```

Now your z/VM system volumes should be relabeled. Verify with the **`QUERY CPOWNED`** command, as in Example 4-42.

Example 4-42 Verifying z/VM volumes with QUERY CPOWNED

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  VVA770  A770  Own    Online and attached
   2  VVA771  A771  Own    Online and attached
   3  VVA772  A772  Own    Online and attached
   4  VVA773  A773  Own    Online and attached
   5  VVA774  A774  Own    Online and attached
   6  VPA775  A775  Own    Online and attached
   7  VPA776  A776  Own    Online and attached
   8  VPA777  A777  Own    Online and attached
   9  VPA778  A778  Own    Online and attached
  10  VPA779  A779  Own    Online and attached
  11  ------  ----  -----  Reserved
  ...
```

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`**.

Important: Only do this if your saved segments have been destroyed! To rebuild saved segments, try the following commands:

```
==> vmfsetup zvm cms
==> sampnss cms
==> i 190 cl parm savesys cms
==> vmfbld ppf segbld esasegs segblist ( all
```

4.11 Restoring your z/VM system from tape

It is good to practice to restore a system periodically. You do not want to perform your first restore when the pressure is on.

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. IPLing one of the two will probably find the correct W01, W02, PAG and SPL volumes, but IPLing the other one will probably find the wrong set.

Perform this step only if you successfully completed 4.9, “Backing up your z/VM system to tape” on page 56, and 4.10, “Relabel the system volumes” on page 57. If you have done both, then the system on tape has volume labels of `520`*xxx* and the system on DASD has volume labels `VV`*yyyy*. You can restore this system to five other 3390-3s.

Refer to the Appendix E “Restore the z/VM System Backup Copy from Tape” in the manual *The z/VM Guide for Automated Installation and Service*, GC204-6099.

5

Servicing z/VM

This section describes how to apply a Programming Temporary Fix (PTF) and a Recommended Service Upgrade (RSU) from envelope files. Both processes are basically the same.

Important: When applying service, there is always a chance that you might want to back it up. It is recommended that you have a backup of your system before starting this section.

The application of corrective service to z/VM is covered in the *z/VM V5R1 Service Guide and 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:

- Installs an RSU or applies CORrective service for z/VM components, features, or products.
- Displays either the RSU level of the component specified or whether a particular PTF or APAR has been applied (when used with STATUS).
- Creates PTF bitmap files (when used with BITMAP).

When **`SERVICE`** successfully completes, the **`PUT2PROD`** EXEC places the z/VM components, features, or products that are installed on the z/VM System DDR into production.

5.1 Applying a PTF

You might 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 can 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 this Web page:

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

So for z/VM 5.2, you want to apply PTF UM31613. Following is an example of how to do that.

Important: It is strongly recommended that you apply this PTF. If you do not, examples of cloning Linux in Chapter 9, “Configure Linux for cloning” on page 131 very likely will fail due to network connectivity problems.

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, as in Example 5-1.

Example 5-1 vmfsetup on MAINT

```
==> 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 Field panel, as in Example 5-2.

Example 5-2 Using VMFINFO to find the ZVM file

```
==> 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 panel.
4. Select **PTFs/APARs** on the VMFINFO Main Panel.
5. Type in the PTF number `UM31613` in the PTF number field then select **Status of PTF** on the *PTF/APAR Queries* panel, as in Example 5-3.

Example 5-3

```
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 in` Example 5-4.

Example 5-4

```
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.

5.1.1 Getting service using Internet FTP

You can get service for z/VM with tapes. However, you might also wish to get 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`

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 **Go**. This should take you to a page entitled Support for VM.
5. Click on **Download specific fixes**. You might 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.
...
```

5.1.2 Downloading the service to z/VM

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

Example 5-5

```
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 lot of free space, so it is a good 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
```

3. Now use the FTP client to get the PTF *envelope files* off 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` as 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. In order 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
```

5.1.3 Receiving, applying, and building service

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

To prepare to use the `SERVICE` command, you must have a 256MB virtual machine and you must have the minidisk with a lot 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 V5.2.0    2006-01-24 13:26
   ==> Enter
   ```

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. Now 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 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: 01/31/06           Time: 12:57:09            ****
   ************************************************************************
   ====> F3
   ```

Ideally there will be no output which as there is in this example. That means the service applied perfectly.

5.1.4 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 as 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 via the **`SHUTDOWN REIPL`** command in order to completely test the changes:

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

Your z/VM system should come back in a few minutes.

5.2 Applying a Recommended Service Upgrade or RSU

Applying an RSU is very similar to applying a PTF described in the previous section. An example of upgrading to an z/VM 5.2 RSU was not available at the time of writing of this book. The example that follows shows an RSU being applied to a z/VM 5.1 system.

z/VM service can be preventive (RSU) or corrective (COR). Part 4, *Service Procedure*, in the manual *Guide for Automated Installation and Service* gives a complete description of applying service to z/VM, however it assumes you are starting with the RSU tape.

The section that follows is a summary of applying service and also describes how to obtain the service via 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 1.0, service level 0401 (64-bit)
Generated at 08/31/04 17:33:32 EST
```

The service level “0401” is split in half. In this example, the “04” means the year 2004, and the “01” means the first service level for that year. In the example that follows the first service level for the year 2005, or “0501” is used.

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

- Make the service available on the Internet via FTP
- Download the service to z/VM
- Receive, apply and build the service
- Put the service into production

5.2.1 Making the service available on the Internet via FTP

The PTF number for the most current RSU for z/VM 5.2.0 is UM97520 The PTF number for the most current RSU for z/VM 5.1.0 is UM97510. 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 on **Support and Downloads** at the top menu.
3. Click on **Downloads and Drivers** on the left frame.
4. Under *Category*, select **zSeries (mainframe)**
5. Under *Operating Systems*, select **z/VM** This should take you to a page entitled Support for VM.
6. Click on **Download selective fixes by PTF**. You may be prompted for your IBM ID and password.
7. In the text box *Enter PTF numbers below [e.g: U412345, U467890]*, enter **UM97520** for the latest z/VM 5.2 service level, **UM97510** for the latest z/VM 5.1service level, or the appropriate PTF number. All other defaults should be correct.
8. Click **Continue**
9. 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.
```

5.2.2 Downloading the service to z/VM

You should receive two e-mails. The first e-mail has your order number. The second e-mail has instructions on 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 rlst8476.bin
TEXT    =     get rptf0176.bin
TEXT    =     get rptf0276.bin
TEXT    =     get rptf0376.bin
```

1. Logon to `MAINT`.
2. The `MAINT 500` disk should have a lot of free space, so it is a good 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
   ```

3. Use the FTP client to get the RSU envelopes off the Internet. The envelope files can be large so this may take some time. It is recommended that you rename the file type from

BIN to SERVLINK via 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 {/390268476/c568411202}
ftp> ascii
ftp> get ftp8476.txt
ftp> binary f 1024
ftp> get rlst8476.bin rlst8476.servlink
...
10240 bytes transferred in 0.523 seconds. Transfer rate 19.58 Kbytes/sec
ftp> get rptf0176.bin rptf0176.servlink
...
36944896 bytes transferred in 191.632 seconds. Transfer rate 192.79 Kbytes/sec.
ftp> get rptf0276.bin rptf0276.servlink
...
26028032 bytes transferred in 132.353 seconds. Transfer rate 196.66 Kbytes/sec.
ftp> get rptf0376.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.

4. The envelope files arrive in a compressed format to speed downloads. In order 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 rlst8476 servlink a = = = (replace
==> deterse rptf0176 servlink a = = = (replace
==> deterse rptf0276 servlink a = = = (replace
==> deterse rptf0376 servlink a = = = (replace
```

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

```
==> browse ftp8476 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 rlst8476.bin byte count is: 10240.
The rptf176.bin byte count is: 36944896.
The rptf276.bin byte count is: 26028032.
The rptf376.bin byte count is: 52193280.
...
```

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.

1. To prepare to use the **SERVICE** command, you must have a 256MB virtual machine and you must have the minidisk with a lot 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 V5.2.0    2006-01-24 13:26
   ==> 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
   ...
   VMFSET2760I VMFSETUP processing completed successfully
   VMFSRV2760I SERVICE processing completed successfully for GCS BUILD
   VMFSUT2760I VMFSUFTB processing started
   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 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, the above the following messages are 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 can lose your connection because the TCP/IP service machine might be recycled. Therefore, you might want to run this command from a console.

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

```
==> 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
```

Even though the service has been “put into production”, the **`QUERY CPLEVEL`** command should still return the current service level; in this example 0401. This is because the new CP load module (nucleus) has not been invoked:

```
==> q cplevel
z/VM Version 5 Release 1.0, service level 0401 (64-bit)
...
```

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 0501:

```
==> shutdown reipl iplparms cons=sysc
...
==> q cplevel
z/VM Version 5 Release 1.0, service level 0501 (64-bit)
...
```

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

You should now be done installing and configuring z/VM. A great attribute of z/VM is that it normally hums along with little maintenance required. It is now time to change your focus to Linux.

6

Configuring an NFS server

It is possible to install Linux onto the mainframe from physical CDs, however, installation over the network from another server using the Network File System (NFS) is recommended. To accomplish this, it is recommended that you set up a PC Linux system. This server will supply both the SLES9 distribution and the files associated with this book.

It must have at least 12 GB of free disk space for one install tree. It will require about 24 GB if you want both an s390 (31-bit) and an s390x (64-bit) trees). It can be a Linux PC, but it can also be a UNIX box (Sun™ Solaris™, Hewlett Packard HP-UX, IBM AIX® or other). The steps in this chapter explain how to configure a PC Linux box as the NFS server.

You can also choose to use a Windows workstation via SMB, but this option is not addressed in this book. Often more problems are encountered when using a Windows workstation to serve the SLES9 install tree so this option is not recommended. If you have no other choice, refer to Section 5.2, “Using a Microsoft Windows Workstation” in the manual *SUSE LINUX Enterprise Server ARCHITECTURE SPECIFIC INFORMATION*. This manual is included on the SuSE CDs.

To get started with Linux on zSeries using this book, you must perform the following tasks:

- 6.1, “Downloading files associated with this book” on page 76
- 6.2, “Setting up an SLES9 install tree” on page 76
- 6.3, “Enabling the NFS server on SLES9” on page 82

6.1 Downloading files associated with this book

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

`ftp://www.redbooks.ibm.com/redbooks/sg246695`

The tar file: `SG24-6695.tgz` is about 55 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 SG24-6695.tgz ...
# tar xzf SG24-6695.tgz
```

List the files in the new directory `virt-cookbook/`:

```
# ls virt-cookbook/
.  ..  README.txt  linux-controller  linux-master  nfs-server  virt-cookbook.pdf  vm
```

You now have downloaded and untarred the files associated with this book.

6.2 Setting up an SLES9 install tree

You may have a licensed version of SLES9 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/linuxenterpriseserver/eval.html`

Follow the link named *SUSE Linux Enterprise Server 9 for IBM zSeries & IBM s/390 Series* and create an account to download the ISO images.

6.2.1 SLES9 ISO image file names and sizes

Table 6-1 lists the file names for the SLES9 31-bit ISO images.

Table 6-1 SLES9 s390 CDs

CD number	File name	File size in bytes
1	SLES-9-s390-RC5a-CD1.iso	320,485,376
2	SLES-9-s390-RC5-CD2.iso	541,788,160
3	SLES-9-s390-RC5-CD3.iso	437,792,768
4	SLES-9-s390-RC5-CD4.iso	498,302,976
5	SLES-9-s390-RC5-CD5.iso	680,095,744
6	SLES-9-s390-RC5-CD6.iso	664,590,336

Table 6-2 lists the file names for the SLES9 64-bit ISO images.

Table 6-2 SLES9 s390x CDs

CD number	File name	File size in bytes
1	SLES-9-s390x-RC5a-CD1.iso	327,061,504
2	SLES-9-s390x-RC5-CD2.iso	604,880,896

CD number	File name	File size in bytes
3	SLES-9-s390x-RC5-CD3.iso	466,620,416
4	SLES-9-s390x-RC5-CD4.iso	509,405,184
5	SLES-9-s390x-RC5-CD5.iso	673,800,192
6	SLES-9-s390x-RC5-CD6.iso	664,592,384

6.2.2 SLES9 Service Pack CDs

You may also have service pack CDs. Service packs are added to the base SLES9 install, however, they are not added to previous service packs. For example, if you want to get to a SLES9 SP3 level, you would use the base CDs and the SP3 CDs, but not the SP1 nor SP2 CDs. In general it is recommended that you use the latest service pack, however, you might want a specific service pack. Therefore, all three service packs are addressed in this section. The file names and sizes of service pack CDs are listed in the tables that follow.

SLES9 SP3 CDs

SLES9 SP3 became available in December of 2005. Table 6-3 lists the CDs for the s390 and s390x distributions:

Attention: After the original SP3 CDs were released in 2005, others were released in 2006 and it is not entirely clear which are the latest and which are the correct file names. On a mirror of the SuSE maintweb site, there appears to be a new CD2 dated Jan. 26, 2006 with the same file name, and a new CD1 dated Feb. 16 with a file name of `CD1a` instead of `CD1`.

The following append to the linux-390 list server from a novell.com e-mail address seems to be the most authoritative:

```
> It looks to me like the latest SP3 CD images for 64-bit are labeled as
> follows:
>
> SLES-9-SP-3-s390x-GM-CD1a.iso  -- e74f074fc81c1d1bb12ff59d9f2541bb
> SLES-9-SP-3-s390x-GM-CD2.iso   -- 9ebbd79d41e46ee3c0f0b4e6997f7319
> SLES-9-SP-3-s390x-RC4a-CD3.iso -- 34ddfb7fc16bd5b6fabb4032766300ad

Yes, these are the correct images for 64-bit mode. Unfortunately, the
files are named a bit inconsistently. BTW, for 31-bit mode:

SLES-9-SP-3-s390-GM-CD1a.iso   b760767a7ee7c8e83d468234f41aca09
SLES-9-SP-3-s390-GM-CD2.iso    cbd1c69ad1192605331fa22d303ae5bc
SLES-9-SP-3-s390-RC4a-CD3.iso  68df79fbb5b0aedff29a330d09d7a817

Regards, Joerg Reuter
```

For this reason, the **`mksles9root.sh`** script has been modified to accept either the file names in the above append, or if they are not found, the file names in the previous table.

Table 6-3 SLES9 SP2 s390 and s390x CDs

CD number	File name	File size in bytes
1	SLES-9-SP-3-s390-GM-CD1.iso	544,542,720 (545,089,536 - CD1a)
2	SLES-9-SP-3-s390-GM-CD2.iso	617,662,464 (617,631,744 - 2nd CD2)
3	SLES-9-SP-3-s390-GM-CD3.iso	663,846,912

CD number	File name	File size in bytes
1	SLES-9-SP-3-s390x-GM-CD1.iso	600,645,632 (601,272,320 - CD1a)
2	SLES-9-SP-3-s390x-GM-CD2.iso	654,403,584 (654,372,864 - 2nd CD2)
3	SLES-9-SP-3-s390x-GM-CD3.iso	662,065,152

SLES9 SP2 CDs

SLES9 SP2 became available in July of 2005. Table 6-4 lists the CDs for the s390 and s390x distributions:

Table 6-4 SLES9 SP2 s390 and s390x CDs

CD number	File name	File size in bytes
1	SLES-9-SP-2-s390-GM-CD1.iso	559,288,320
2	SLES-9-SP-2-s390-GM-CD2.iso	569,640,960
3	SLES-9-SP-2-s390-GM-CD3.iso	576,258,048
1	SLES-9-SP-2-s390x-GM-CD1.iso	612,798,464
2	SLES-9-SP-2-s390x-GM-CD2.iso	609,878,016
3	SLES-9-SP-2-s390x-GM-CD3.iso	573,409,280

SLES9 SP1 CDs

SLES9 SP1 became available in January of 2005. Table 6-5 lists the CDs for the s390 and s390x distributions:

Table 6-5 SLES9 SP1 s390 and s390x CDs

CD number	File name	File size in bytes
1	SLES-9-SP-1-s390-RC5-CD1.iso	581,201,920
2	SLES-9-SP-1-s390-RC5-CD2.iso	642,019,328
1	SLES-9-SP-1-s390x-RC5-CD1.iso	713,717,760
2	SLES-9-SP-1-s390x-RC5-CD2.iso	689,106,944

6.2.3 Starting from physical CDs

If you are starting with physical CDs, you must first convert them to ISO images. This can be accomplished with the Linux **dd** command, which basically performs a byte-for-byte copy of the CD contents.

Put the first CD in the drive. It is often available as the file `/dev/cdrom`. If there is no such file on your system, you must determine which file (such as `/dev/hdc`) is the device file for the CD drive. Before copying the CDs, it is recommended that you create two directories to keep the 31-bit and 64-bit distributions separate:

```
# cd /nfs
# mkdir sles9 sles9x
```

The sections that follow assume you are installing a 64-bit SLES9 SP3 system, so file names typically contain sles9x and sp3. If you are installing a 31-bit system, omit the x where appropriate. If you are installing no service pack or a service pack other than 3, specify the file names accordingly.

Now copy the contents of the first CD to an ISO image using the **`dd if`** (input file) and **`of`** (output file) parameters. The following example copies the first 64-bit SLES9 CD to the appropriately named file. (**Note**: the file names must be identical to those in the preceding tables for the **`mksles9root.sh`** script to succeed.) Sometimes, `/dev/cdrom` is automatically mounted over `/mnt/cdrom` when you put the CD in the drive. If so, you must unmount it with the **`umount`** command after copying the contents of the CD:

```
# cd sles9x
# dd if=/dev/cdrom of=SLES-9-s390x-RC5a-CD1.iso
# umount /mnt/cdrom
```

The CD should start spinning and this will take a couple of minutes to copy. Remove the CD and insert the next. Repeat the **`dd`** command for each of the physical CDs you have, using the file names shown in the preceding tables.

6.2.4 Verifying the ISO images

Before you create an install tree, you should verify the integrity of the ISO images. This is done using a file of checksum values and ISO file names. The checksums were calculated from the contents of the CD. After downloading, **`dd`**ing, the ISO images, the checksums are calculated again and compared against the original values with the **`md5sum`** command and the checksum files. For convenience, the checksum files are included in the `nfs-server/` directory (Example 6-1).

Example 6-1 Checksum file list

```
# ls /nfs/virt-cookbook/nfs-server/MD5SUMS.*
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9x
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp1
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp1
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp2
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp2
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp3
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp3
```

SLES9 MD5SUM files

For reference, the contents of the 64-bit and 31-bit SLES9 MD5SUM files are listed in Example 6-2 and Example 6-3.

```
# for i in /nfs/virt-cookbook/nfs-server/MD5SUMS.*; do echo $i; cat $i; echo; done
```

Example 6-2 31-bit MD5SUM values

```
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9
4f5ca784a148ac0431ee18cd9c7840ce  SLES-9-s390-RC5a-CD1.iso
b7f95b81510dfd527579b184baee5d30  SLES-9-s390-RC5-CD2.iso
21d3e2eb32aa83bb0ddad1f08526d37f  SLES-9-s390-RC5-CD3.iso
849936436e1fdd351df6403dbf5d7e2d  SLES-9-s390-RC5-CD4.iso
7655c5871fae2915efe9644d211041e4  SLES-9-s390-RC5-CD5.iso
11bb0fcd7e82a07b7ab2878708478768  SLES-9-s390-RC5-CD6.iso

/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp1
a6c0e172ac05b5ffbe29de7beb8ca3dc  SLES-9-SP-1-s390-RC5-CD1.iso
b9446d21dcc1b18694626979ccef3df9  SLES-9-SP-1-s390-RC5-CD2.iso
```

```
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp2
0a03c112f68b8167e64fd882e67c12ea  SLES-9-SP-2-s390-GM-CD1.iso
016b027f69b919347a7f1929eec03847  SLES-9-SP-2-s390-GM-CD2.iso
684ecbd9866a1f925d6e77b96d2bc5cc  SLES-9-SP-2-s390-GM-CD3.iso

/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9sp3
476846b96e4f2a6bcb5a6a612964199e  SLES-9-SP-3-s390-GM-CD1.iso
8cd66bf243fcb0123011240b26a0682b  SLES-9-SP-3-s390-GM-CD2.iso
68df79fbb5b0aedff29a330d09d7a817  SLES-9-SP-3-s390-GM-CD3.iso
```

Example 6-3   64-bit MID5SUM values

```
/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9x
b45dd5dbb47fad87f58fb1b658dbc7f8  SLES-9-s390x-RC5a-CD1.iso
04e7b9d8629e1d973230be67b22dc9b7  SLES-9-s390x-RC5-CD2.iso
1a5b5f0d15b4e182d2b23da478725933  SLES-9-s390x-RC5-CD3.iso
cee521cbd1ac6c2e3980da99bd4b652e  SLES-9-s390x-RC5-CD4.iso
7fd515da7f5cdc2bd916ee432da30fd4  SLES-9-s390x-RC5-CD5.iso
74301159228f91a4da4a2bb78cb607aa  SLES-9-s390x-RC5-CD6.iso

/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp1
897e8b4f082a9606f92fb54302ee6638  SLES-9-SP-1-s390x-RC5-CD1.iso
74541daa9e4fa05ec5d3f45909c73dd5  SLES-9-SP-1-s390x-RC5-CD2.iso

/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp2
77cf7413affb85badd9037b4b62c1e65  SLES-9-SP-2-s390x-GM-CD1.iso
86c63b8c99e4f3c09bc2b9b81d8bdf1a  SLES-9-SP-2-s390x-GM-CD2.iso
22e4c3cf160910c2913c07e8ccc322eb  SLES-9-SP-2-s390x-GM-CD3.iso

/nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp3
8fd3e6d2f2cedfd0a079994101b491d9  SLES-9-SP-3-s390x-GM-CD1.iso
9072e771ad4412697faba538b89f9de2  SLES-9-SP-3-s390x-GM-CD2.iso
34ddfb7fc16bd5b6fabb4032766300ad  SLES-9-SP-3-s390x-GM-CD3.iso
```

Use the **`md5sum -c`** command to verify the integrity of the ISO images. All should report OK. Example 6-4 shows how to verify the six 64-bit SLES9 and three 64-bit SP3 ISO images:

Example 6-4   Using md5sum -c to verify files

```
# cd /nfs/sles9x
# md5sum -c /nfs/virt-cookbook/nfs-server/MD5SUMS.sles9x
SLES-9-s390x-RC5a-CD1.iso: OK
SLES-9-s390x-RC5-CD2.iso: OK
SLES-9-s390x-RC5-CD3.iso: OK
SLES-9-s390x-RC5-CD4.iso: OK
SLES-9-s390x-RC5-CD5.iso: OK
SLES-9-s390x-RC5-CD6.iso: OK
# md5sum -c /nfs/virt-cookbook/nfs-server/MD5SUMS.sles9xsp3
SLES-9-SP-3-s390x-GM-CD1.iso: OK
SLES-9-SP-3-s390x-GM-CD2.iso: OK
SLES-9-SP-3-s390x-GM-CD3.iso: OK
```

Any ISO images that do not report `OK` must be downloaded or copied again.

6.2.5 Creating the SLES9 install tree

Now that you have created and verified the appropriate ISO images, you can create a SLES9 install tree for either the 31-bit (s390) or 64-bit (s390x) distributions. The SuSE SLES9 manual documents how to create an install tree, however, the process can become cumbersome and is not documented in the clearest fashion when Service Packs are included. Further, there appears to be a bug in applying service packs 2 and 3 with the YaST => Software => Patch CD Update process whereby updated packages on CD2 cannot be read.

For convenience, and to work around the apparent Patch CD Update bug, a script, **`mksles9root.sh`**, is available. The source code for this script is in A.6.1, "The mksles9root.sh script" on page 221. It takes one argument which must be either **`s390`** to create a 31-bit tree or **`s390x`** for a 64-bit tree. The script looks for ISO images with the file names listed and create a directory tree that can be used for both installing and upgrading older SLES9 systems to SP2 or SP3.

The **`mksles9root.sh`** script must be run from the directory with the ISO files to create the SLES9 install tree. This step should take about five to eight minutes. The output for a 64-bit distribution when the SP3 ISO images are found is shown in Example 6-5.

Example 6-5 mksles9root.

```
# cd /nfs/sles9x
# /nfs/virt-cookbook/nfs-server/mksles9root.sh s390x
Making a SLES9 install tree ...
SP3 ISO images found ...
The tree named sles9xsp3root/ will be SLES9 + SP3 ...
Making the directory structure ...
Copying SLES9 ISO images ...
  Mounting and copying SLES-9-s390x-RC5a-CD1.iso ...
  Mounting and copying SLES-9-s390x-RC5-CD2.iso ...
  Mounting and copying SLES-9-s390x-RC5-CD3.iso ...
  Mounting and copying SLES-9-s390x-RC5-CD4.iso ...
  Mounting and copying SLES-9-s390x-RC5-CD5.iso ...
  Mounting and copying SLES-9-s390x-RC5-CD6.iso ...
Copying SLES9 SP3 ISO images ...
  Mounting and copying SLES-9-SP-3-s390x-RC4a-CD1.iso ...
  Mounting and copying SLES-9-SP-3-s390x-RC4a-CD2.iso ...
  Mounting and copying SLES-9-SP-3-s390x-RC4a-CD3.iso ...
Removing temporary mount point ...
Making symbolic links ...
Creating yast/*order files ...
  Creating files for SLES9 + SP3...
Merging patches for service pack 3 ...
Hacking mediamap file for service pack 3 ...
The install tree is built under sles9xsp3root/
```

After the script is completed, the SLES9 install tree is populated under a new directory which is named one of the following eight names in Example 6-6.

Example 6-6 SLES9 64-bit directory names

```
sles9xroot/SLES9 64-bit
sles9xsp3root/SLES9 64-bit + SP3
sles9xsp2root/SLES9 64-bit + SP2
sles9xsp1root/SLES9 64-bit + SP1
sles9root/SLES9 31-bit
sles9xsp3root/SLES9 31-bit + SP3
sles9xsp2root/SLES9 31-bit + SP2
sles9xsp1root/SLES9 31-bit + SP1
```

The directory created from the previous example is `sles9xsp3root/`. You can choose to repeat this process and build both 31-bit and 64-bit trees. If so, be sure you have enough disk space. However, another alternative is to build one tree, move it over to zSeries Linux, delete it on the NFS server and repeat the process for the second distribution.

You should now have a directory, `/nfs/`, with three subdirectories, as in Example 6-7.

Example 6-7 /nfs/ subdirectories

```
/nfs/
|-- sles9/
|-- sles9x/
    |-- sles9xsp3root/
`-- virt-cookbook/
    |-- linux-controller
    |-- linux-master
    |-- nfs-server
    `-- vm
```

You should have populated either `/nfs/sles9/` for 31-bit or `/nfs/sles9x/` for a 64-bit, or perhaps both. You should also have populated the `virt-cookbook/` directory with files associated with this book. The next step is to enable the NFS server.

6.3 Enabling the NFS server on SLES9

Your method of enabling an NFS server will differ from ours, depending upon the operating system. However, the steps are basically the same:

1. Verify that you have NFS RPMs installed.
2. Export the appropriate directories.
3. Start the NFS server in the current run level.

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 with 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, as in Example 6-8.

Example 6-8 Copying and modifying the exports.orig file

```
# cd /etc
# cp exports exports.orig
# vi exports       // add two lines at the bottom
/nfs/virt-cookbook             *(ro,sync)
/nfs/sles9x/sles9xsp3root      *(ro,sync)
```

The `*(ro,sync)` parameter specifies that any client with access to this server can get the NFS mount read-only. You might want to be more restrictive than any client (`*`) for security reasons. Type `man exports` for more details.

Be sure the NFS server is running in your run level. For a SLES Linux, the service name is `nfsserver`. This can be accomplished with the **`chkconfig -l`** command:

```
# chkconfig -l nfsserver
nfsserver                0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

This 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 must set it to run with the **`chkconfig`** command and turn it on for the current run level with the **`rcnfsserver start`** command:

```
# chkconfig nfsserver on
# rcnfsserver start
Starting kernel based NFS server
done
```

For RHEL4: To start the NFS server on Red Hat RHEL4, the parameter to **`chkconfig`** is **`nfs`**, not `nfsserver`. Also, the **`service`** command is used to start and stop services rather than the rc* symbolic links:

```
# chkconfig nfs on
# chkconfig --list nfs
nfs       0:off  1:off  2:off  3:on   4:off  5:on   6:off
# service nfs start
Starting NFS services:                                [  OK ]
Starting NFS quotas:                                  [  OK  ]
Starting NFS daemon:                                  [  OK  ]
Starting NFS mountd:                                  [  OK  ]
```

Your NFS server should now be running with the directory exported. It is recommended that you test this by mounting the exported directory locally. Example 6-9 shows that the `/mnt/` directory is empty. The newly exported `/nfs/` directory is mounted and the files are listed.

Example 6-9 Exporting and mounting the /nfs/ directory

```
# ls /mnt
# mount -t nfs localhost:/nfs/sles9x/sles9xsp3root /mnt
# ls -F /mnt
boot@     control.xml@  driverupdate@  media.1@  sles9/  yast/
content@  core9/        linux@         s390x/    sp3-9/
```

This shows that the SLES9 install root directory is accessible. Now unmount it and test the virt-cookbook directory, as in Example 6-10.

Example 6-10 Testing the virt-cookbook

```
# umount /mnt
# mount -t nfs localhost:/nfs/virt-cookbook /mnt
# ls -F /mnt
linux-controller/  linux-master/  nfs-server/  README.txt  virt-cookbook.pdf  vm/
# umount /mnt
```

You should now 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 in on zSeries Linux.

7

Installing and configuring Linux

Chapters 4, 5 and 6 must be completed before proceeding with this and following chapters.

By now, you should have created a new z/VM user ID, `LNXMAINT`. Now it is time to create the first Linux user ID, `SLES9X`. This Linux ID is unique in that it will have two copies of Linux; think of it as a dual-boot Linux PC. You will install the following Linux images onto `SLES9X`:

Master image	The copy of Linux that will be cloned. This should be as lean as possible so as to be a generic virtual server and to fit comfortably in one 3390-3 DASD.
Controller	The copy of Linux that will normally be running and one that does the cloning.

In addition to being the cloner, the controller can have other functions:

- SLES9 install server - a tree of the RPMs and other files needed for installation
- NFS server - to export the *install tree* and possibly other data and directories
- Backup server - for incremental backup of key virtual server configuration files
- Time server - for time synchronization of all other Linux virtual servers

In this chapter, you perform the following tasks:

7.1 Creating the user ID SLES9

In this section you define the `SLES9X` user ID to z/VM.

1. **Logon to** `MAINT` and 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 in the `INCLUDE` statement. You used the existing profile `TCPCMSU` when you defined the `LNXMAINT` user.

2. 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 38-49:

Example 7-1 Creating the LNXDFLT profile

```
""    *
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
""046   LINK MAINT 019E 019E RR
```

3. Edit the duplicated profile and add some lines and insert the emboldened text (Example 7-2).

Example 7-2 Editing the duplicate file

```
PROFILE LNXDFLT
  IPL CMS                             1
  MACHINE ESA 4                       2
  CPU 00 BASE                         3
  CPU 01                              4
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1    5
  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            6
  LINK TCPMAINT 592 592 RR            7
```

Notes:

1 CMS will be IPLed when the user ID is logged on.
2 Machine will of type ESA with a maximum of 4 CPUs that can be defined.
3 Defines the base CPU.
4 Defines a second CPU. *Do not include this* if your LPAR has only a single IFL/CP.
5 Defines a virtual NIC connected to the VSWITCH starting at virtual address 600.
6 Provides read access to `LNXMAINT 192 disk` as the user's 191 disk.
7 Provides read access to `TCPMAINT 592 disk`, so that the user has access to TCP/IP services such as FTP.

4. Go to the bottom of the file and add the definition for a new user ID named `SLES9X`. This user ID is given class B privilege, aside from the typical class G, in order to run the **`FLASHCOPY`** command and class E privilege to run the **`QUERY NSS`** command. Be sure to replace the volume labels (`VMA781-VMA786` in Example 7-3) with the labels of your DASD.

Example 7-3 Adding SLES9X

```
USER SLES9X LNX4VM 256M 1G BEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3038 <VMA781> MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 <VMA781> MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
 MDISK 103 3390 0001 3338 <VMA782> MR LNX4VM LNX4VM LNX4VM
 MDISK 104 3390 0321 3018 <VMA783> MR LNX4VM LNX4VM LNX4VM
 MDISK 105 3390 0001 3338 <VMA784> MR LNX4VM LNX4VM LNX4VM
 MDISK 106 3390 0001 3338 <VMA785> MR LNX4VM LNX4VM LNX4VM
 MDISK 107 3390 0001 3338 <VMA786> MR LNX4VM LNX4VM LNX4VM
*
```

This Linux user ID will have the following minidisks:

100	The root file system of the master image - this is the copy that will be cloned.
101	This is a VDISK swap space created by **`SWAPGEN`** upon logging on. It is NOT defined in `USER DIRECT`, but defined in the user’s **`PROFILE EXEC`** so that when the user ID logs on the VDISK is created.
102	A secondary 300 cylinder minidisk swap space.
103	The root file system of the Linux controller. This will serve as the administration point for all your Linux virtual servers.
104	A 3018 cylinder minidisk for a file system to be mounted over `/backup/` for rsync backup purposes.
105-108	Minidisks used to create a logical volume mounted over `/nfs/` for making the SLES9 64-bit installation tree and the files associated with this book available via NFS.

5. Go back to the top of the file and search for string `USER $ALLOC$`. Add cylinder 0 of each of the six new volumes to this dummy user ID (Example 7-4), so they do not show up as gaps.

Example 7-4 Adding cylinder 000 to the dummy user ID

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
  MDISK A01 3390 000 001 VVA770 R
  MDISK A02 3390 000 001 VVA773 R
  MDISK A03 3390 000 001 VVA774 R
  MDISK A04 3390 000 001 VMA781 R
  MDISK A05 3390 000 001 VMA782 R
  MDISK A06 3390 000 001 VMA783 R
  MDISK A07 3390 000 001 VMA784 R
  MDISK A08 3390 000 001 VMA785 R
  MDISK A09 3390 000 001 VMA786 R
...
====> file
```

6. Run **`DISKMAP`** to check for overlaps and gaps. You should only see the single 501 cylinder gap (Example 7-5).

Example 7-5 Checking for gaps and overlaps

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
--------------------  4  line(s) not displayed --------------------
                                        0         500         501    GAP
--------------------  322  line(s) not displayed --------------------
====> quit
```

7. When the disk layout is correct, run **`DIRECTXA`** to bring the changes online:

   ```
   ==> directxa user
   z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
   EOJ DIRECTORY UPDATED AND ON LINE
   ```

You have now defined the user ID that will be both the master Linux image and the controller.

7.2 Preparing SLES9 bootstrap files

To IPL a SLES9 installation system, three bootstrap files must be prepared, punched to, and PLed from the reader (virtual address `00C`). These 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. The `SLES9X` parameter file and `SLES9X EXEC` were already moved to `LNXMAINT` in 4.6.5, “Copying files associated with this book to LNXMAINT” on page 52. Therefore, only the kernel and RAMdisk need to be copied.

1. Start an SSH session as root on the NFS server.
2. Use the **`ftp`** command to copy the SLES9X kernel and initial RAMdisk to `LNXMAINT`’s `D` disk. These files must have a record format of fixed 80 byte records. This format can be set with the **`site fix 80`** FTP subcommand (if this subcommand fails, try **`quote site fix 80`**). Example 7-6 illustrates this.

Example 7-6 Using ftp to copy the SLES9X

```
# cd /nfs/sles9x/sles9xsp3root/boot>
# ftp <129.40.178.124>
220-FTPSERVE IBM VM Level 520 at LAT124.PBM.IHOST.COM, 15:46:31 EST WEDNESDAY 2004-12-08
220 Connection will close if idle for more than 5 minutes.
Name (129.40.178.124:root): lnxmaint
331 Send password please.
Password:
ftp> cd lnxmaint.192
230 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 SLES9X.KERNEL
local: vmrdr.ikr remote: SLES9X.KERNEL
...
ftp> put initrd SLES9X.INITRD
local: initrd remote: SLES9X.INITRD
...
ftp> quit
```

3. Go back to your 3270 session. **Logoff of MAINT and logon to LNXMAINT**.
4. The files `SLES9X EXEC` and `SLES9X PARMFILE` should exist on the `LNXMAINT 192` disk (D) because we copied them in 4.6.5, “Copying files associated with this book to LNXMAINT” on page 52. Use the **`FILELIST`** command to verify that they were copied in Fixed 80-byte record format. You should see the following files in Example 7-7. The number of records might vary.

Example 7-7 Verifying SLES9X EXEC and SLES9X PARMFILE are on LINXMAINT

```
==> filel sles9x * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=4 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SLES9X   INITRD   D1 F         80     162128       3167  1/25/06 12:11:59
      SLES9X   KERNEL   D1 F         80      59299        784  1/25/06 12:08:45
      SLES9X   EXEC     D1 V         74          9          1  1/24/06 14:48:15
      SLES9X   PARMFILE D1 V         62          8          1  1/24/06 14:48:15
```

5. Quit by pressing **F3**.
6. Verify that the file **`SLES9X EXEC`** has the correct information. Note the kernel and RAMdisk have hard coded file names, but the file name of the parameter file will be the user ID (**`userid()`** function) of the user running the EXEC (Example 7-8).

Example 7-8 Veryfying SLES9X EXEC

```
==> type sles9x exec d
/* EXEC to punch SLES9 install system to reader and IPL from it */
'CP SPOOL PUN *'
'CP CLOSE RDR'
'PUR RDR ALL'
'PUN SLES9X       KERNEL * (NOH'
'PUN' userid() 'PARMFILE * (NOH'
'PUN SLES9X       INITRD * (NOH'
'CH RDR ALL KEEP'
'IPL 00C CLEAR'
```

7. Edit the file `SLES9X PARMFILE`. The fields you should change are in **`<bold>`**. Refer to the worksheet in section 2.5.3, “Linux resources worksheet” on page 16.

Example 7-9 Editing SLES9X PARMFILE

```
==> x sles9x parmfile d
```

Before editng:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=xxxxxx IP_ADDR=n.n.n.n AUTOINSTALL=yes
IP_HOST=xxxxxx.xxx.xxx IP_GATEWAY=n.n.n.n
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=255.255.255.0
IP_BROADCAST=n.n.n.n READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=n.n.n.n
INST_INFO=nfs INST_IP_ADDR=n.n.n.n
INST_IP_DIR=/nfs/sles9x/sles9xsp3root INST_SCREEN=VNC VNC_PASSWORD=xxxxxx
```

After editing:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=<lnx4vm> IP_ADDR=<129.40.178.127> AUTOINSTALL=yes
IP_HOST=<lat127.pbm.ihost.com> IP_GATEWAY=<129.40.178.254>
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=<255.255.255.0>
IP_BROADCAST=<129.40.178.255> READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=<129.40.106.1>
INST_INFO=nfs INST_IP_ADDR=<129.40.46.206>
```

```
INST_IP_DIR=/nfs/sles9x/sles9xsp3root INST_SCREEN=VNC VNC_PASSWORD=<lnx4vm>
```

8. Save your changes with the `FILE` subcommand.

Now you are ready to start the master image installation.

7.3 Installing the master image

The master image is installed onto the `SLES9X 100-102` disks in this section.

Pick up a `PROFILE EXEC` from `LNXMAINT 192` that will be run when you logon to `SLES9X`. It will create a VDISK with the `SWAPGEN EXEC` to be used as an in-memory swap space and will prompt you to IPL Linux. is being picked up from the `LNXMAINT 192` disk. Example 7-10 shows the contents of this EXEC.

Example 7-10 PROFILE EXEC contents

```
==> type profile exec d
/* PROFILE EXEC for the z/VM and Linux Virtualization Cookbook */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 101 524288' /* create a 256M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (id = 'LINUX00' | id = 'LINUX000') then /* this is the controller */
  iplDisk = 103
else /* this is a Linux appliance */
  iplDisk = 100
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL' iplDisk
else  /* user is interactive -> prompt */
do
  say 'Do you want to IPL Linux from DASD' iplDisk'? y/n'
  parse upper pull answer .
  if (answer = 'Y') then
    'CP IPL' iplDisk
end /* else */
```

1. **Logoff of LNXMAINT and logon to SLES9X**. When you logon, you should see a message indicating that a virtual NIC has been created at address 0600 and that a VDISK 101 has been created (Example 7-11).

Example 7-11 NIC 0600 created

```
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0401 (64-bit),
...
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)

LOGON SLES9X
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0501 (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 12:33:30 EST WEDNESDAY 01/25/06
z/VM V5.2.0    2006-01-24 13:26
```

```
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 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 DASD 103? y/n
==> n
```

3. Before you install Linux, it is good to verify the resources. Verify that you have DASD (minidisks) at virtual addresses 100-102 with the **`QUERY VIRTUAL`** command (other Linux IDs without class B privilege will just use the **`QUERY`** command):

```
==> q v 100-102
00: DASD 0100 3390 VMA781 R/W       3038 CYL ON DASD  A781 SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 0017
00: DASD 0102 3390 VMA781 R/W        300 CYL ON DASD  A781 SUBCHANNEL = 0001
```

4. Verify that you have a virtual OSA at addresses 600-602 with the **`QUERY VIRTUAL OSA`** command (Example 7-12).

Example 7-12 Verifying virtual OSA

```
==> q v osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0008
00:      0600 DEVTYPE OSA        CHPID 10 OSD
00:      0600 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
00: OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0009
00:      0601 DEVTYPE OSA        CHPID 10 OSD
00:      0601 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
00: OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 000A
00:      0602 DEVTYPE OSA        CHPID 10 OSD
00:      0602 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
```

5. Use the **`QUERY VIRTUAL STORAGE`** command to show that you have a 256MB machine:

```
==> q v stor
00: STORAGE = 256M
```

6. This is adequate memory to run a SLES9X Linux image, however it is often too small with which to install Linux. Temporarily modify the storage up to 512MB with the **`DEFINE STORAGE`** command. Then **`IPL CMS`** and again answer **n** to the question of IPLing Linux (Example 7-13).

Example 7-13 Temporarily modifying storage for installtion of SLES9X

```
==> def stor 512m
00: STORAGE = 512M
00: Storage cleared - system reset.
==> ipl cms
z/VM V5.2.0    2005-12-22 09:36

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
Do you want to IPL Linux from DASD 103? y/n
n
```

7. Verify that you have a 512MB virtual machine:

```
==> q v stor
00: STORAGE = 512M
```

This change is for the duration of the user ID session. When you logoff and log back on this user ID, the storage will go back to 256MB.

7.3.1 Begin the SLES9 installation

Follow these steps to begin the installation of SLES9.

1. Run the **`SLES9X EXEC`**. You should see many scrolling screens of questions and answers. If you had used the default parameter file shipped with SLES9, you would have had to answer all the networking questions manually. With the proper parameters set in `SLES9X PARMFILE`, the install process should proceed to where you have to use a browser to VNC client get into the YaST2 installation program:

Example 7-14 Running SLES9X EXEC

```
==> sles9x
00: 0000003 FILES CHANGED
Linux version 2.6.5-7.97-s390 (geeko@buildhost) (gcc version 3.3.3 (SuSE Linux))
 #1 SMP Fri Jul 2 14:21:59 UTC 2004
We are running under VM (31 bit mode)
This machine has an IEEE fpu
...
```

Important: If you see the following error that the gateway cannot be **`ping`**ed:

```
HCPIPN2833E IP address is already registered....
...
Warning: The gateway address 129.40.178.254 did not ping.Do you want to ignore this
error and continue anyway? (Yes/No)
```

The reason for it might be a bug in the OSA Express card. If so, it might be due to a race condition. You can try working around this error by IPLing the system again from the reader with the command and you might have to try multiple times.

```
==> #CP IPL 00C
```

This bug was fixed with OSA microcode level 6.26. It is in MCL07 at driver 55 EC J13477 Driver 55 is for the z/890 and z/990. The OSD 626 code level was available on Feb. 16th 2005. Driver 3G is for the z/800 and z/900. The latest code level for the OSD code level 35A was available on March 2nd 2005.

You should see the message:

```
*** You can connect to 129.40.178.127, display :1 now with vncviewer
*** Or use a Java capable browser on http://129.40.178.127:5801/
```

2. 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 93 shows VNC access through a Java-enabled browser. In addition to a browser, you can also use a standalone VNC client if desired.

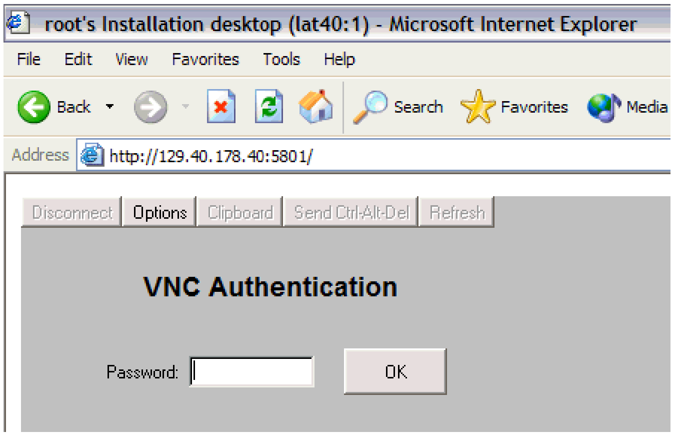

root's Installation desktop (lat40:1) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://129.40.178.40:5801/
Disconnect
Options
Clipboard
Send Ctrl-Alt-Del
Refresh
VNC Authentication
Password:
OK

Figure 7-1 VNC viewer through a Java-enabled browser

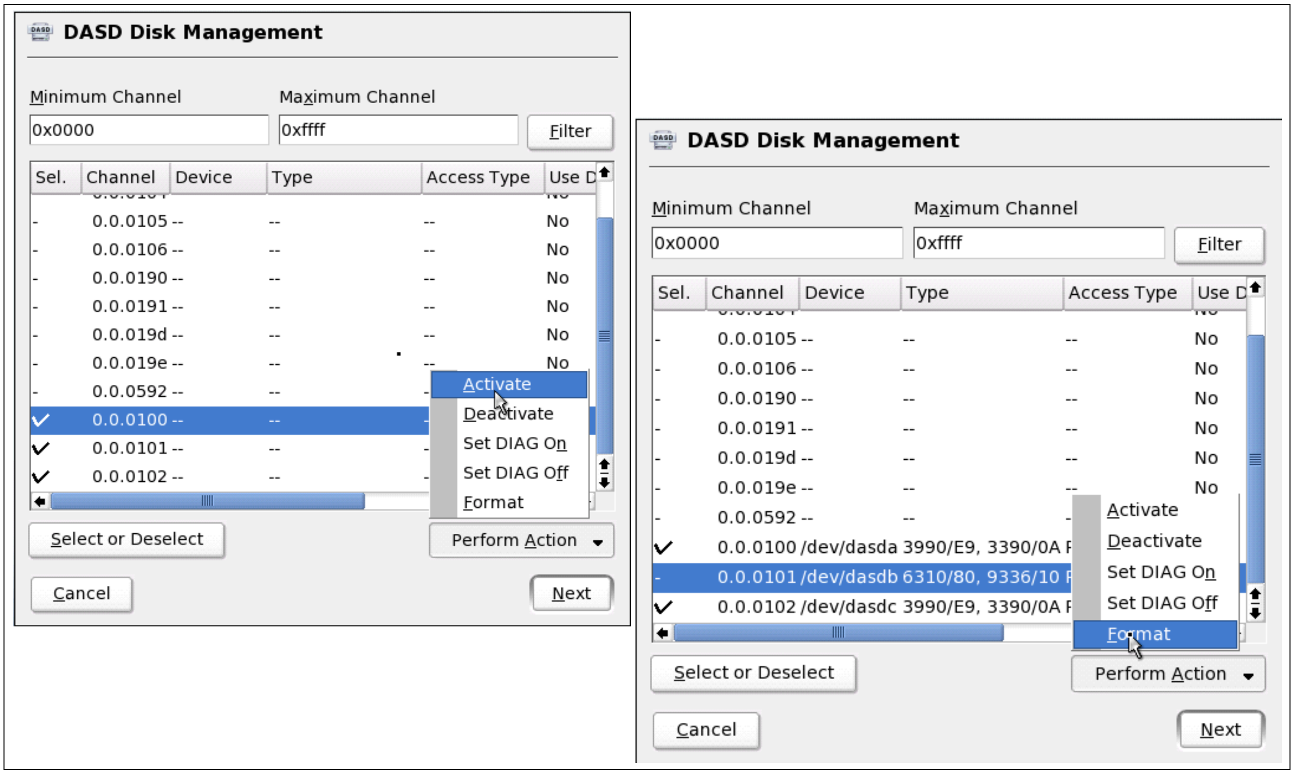
3. You can disconnect from the 3270 session and messages to the console will be lost. If you stay connected, you must clear the screen periodically, or the install process might be delayed waiting for the screen to clear itself.
4. Enter the URL in your browser and log in using the `VNC_PASSWORD` that was specified in `SLES9X PARMFILE`.

Now the installation process should begin.

7.3.2 Beginning YaST installation

The installation program that is running is **`yast2`**. Perform the following steps:

1. The SLES9 license agreement window should open. Click **I agree**.
2. Choose the language, **English**, (or your language) and click **Accept**.
3. The YaST2 DASD Disk Management window should open. You can see all the DASD available to `SLES9X`.
   a. Use 100-102 for the master image.
   b. Highlight each of the three channels and click **Select** or **Deselect** to select them.
   c. You should see a check mark next to those you just selected. Activate them for the Linux you are about to install. Click **Perform Action** → **Activate**. as shown in the left side of Figure 7-2 on page 94.
   d. Disks 100 and 102 must be formatted so that Linux can use them. Deselect disk 101, so that 100 and 102 remain selected. Now click **Perform Action** → **Format** as shown on the right side of the figure.



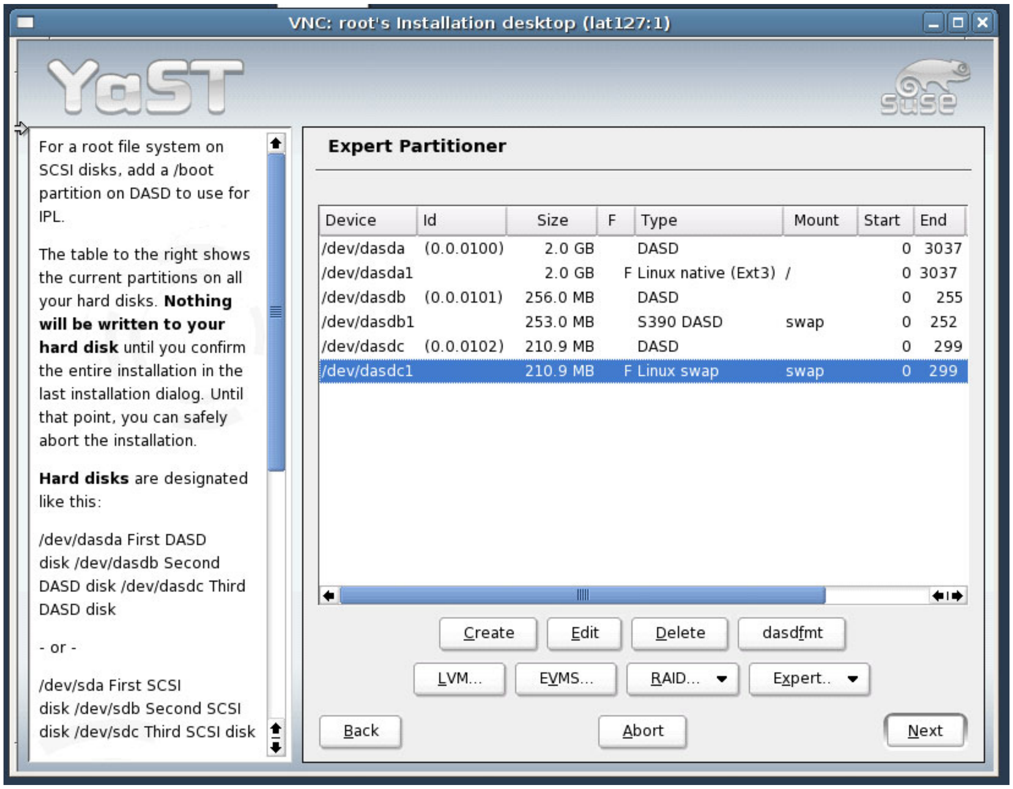
Minimum Channel: 0x0000
Maximum Channel: 0xffff

Sel.	Channel	Device	Type	Access Type	Use D
-	0.0.0105	--	--	--	No
-	0.0.0106	--	--	--	No
-	0.0.0190	--	--	--	No
-	0.0.0191	--	--	--	No
-	0.0.019d	--	--	--	No
-	0.0.019e	--	--	--	No
-	0.0.0592	--	--	-	
✓	0.0.0100	--	--	-	
✓	0.0.0101	--	--	-	
✓	0.0.0102	--	--	-	

Sel.	Channel	Device	Type	Access Type	Use D
-	0.0.0105	--	--	--	No
-	0.0.0106	--	--	--	No
-	0.0.0190	--	--	--	No
-	0.0.0191	--	--	--	No
-	0.0.019d	--	--	--	No
-	0.0.019e	--	--	--	No
-	0.0.0592	--	--	-	
✓	0.0.0100	/dev/dasda	3990/E9, 3390/0A		
-	0.0.0101	/dev/dasdb	6310/80, 9336/10		
✓	0.0.0102	/dev/dasdc	3990/E9, 3390/0A		

Figure 7-2 DASD that are available to SLES9X

4. A window asking for two Parallel Formatted Disks opens. Click **OK**.
5. Click **Yes** to the question Really format the following?
6. A progress indicator window opens. This step can take five to fifteen minutes, depending on the type of channel and the speed of the disks.
7. When the formatting is complete, click **Next**.
8. A window opens asking for the type of installation. Select **New installation** and click **OK**.
9. This brings you to the Installation Settings window. Click **Partitioning**. The Expert Partitioner window opens as shown in Figure 7-3 on page 95.
10. Highlight **/dev/dasda1** and click **Edit**.



For a root file system on SCSI disks, add a /boot partition on DASD to use for IPL.

The table to the right shows the current partitions on all your hard disks. **Nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

Hard disks are designated like this:

/dev/dasda First DASD disk /dev/dasdb Second DASD disk /dev/dasdc Third DASD disk

- or -

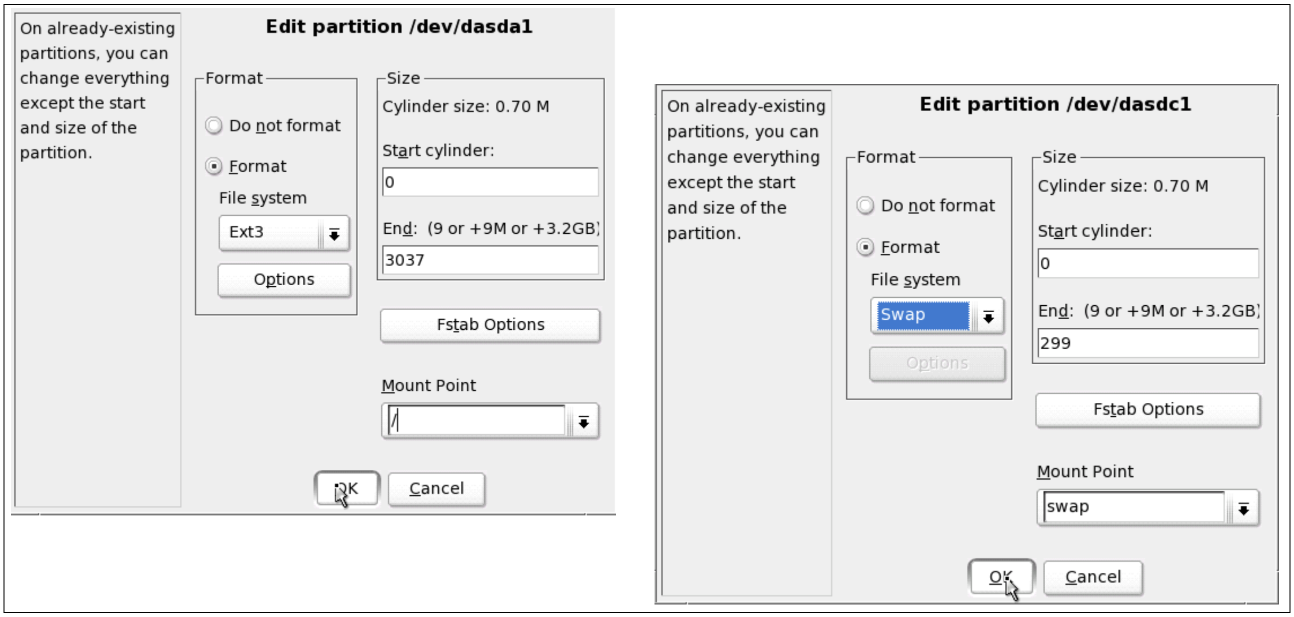
/dev/sda First SCSI disk /dev/sdb Second SCSI disk /dev/sdc Third SCSI disk

Expert Partitioner

Device	Id	Size	F	Type	Mount	Start	End
/dev/dasda	(0.0.0100)	2.0 GB		DASD		0	3037
/dev/dasda1		2.0 GB	F	Linux native (Ext3)	/	0	3037
/dev/dasdb	(0.0.0101)	256.0 MB		DASD		0	255
/dev/dasdb1		253.0 MB		S390 DASD	swap	0	252
/dev/dasdc	(0.0.0102)	210.9 MB		DASD		0	299
/dev/dasdc1		210.9 MB	F	Linux swap	swap	0	299

Figure 7-3 Disk partitioner - before customization

11. You should see a window similar to that shown in the left side of Figure 7-4.
    a. Click the **Format** radio button.
    b. Choose **Ext3** as the file system
    c. Select a mount point of **/**.
    d. Click **OK**.



On already-existing partitions, you can change everything except the start and size of the partition.

Edit partition /dev/dasda1

Format
○ Do not format
● Format
File system: Ext3

Size
Cylinder size: 0.70 M
Start cylinder: 0
End: (9 or +9M or +3.2GB) 3037

Mount Point: /

On already-existing partitions, you can change everything except the start and size of the partition.

Edit partition /dev/dasdc1

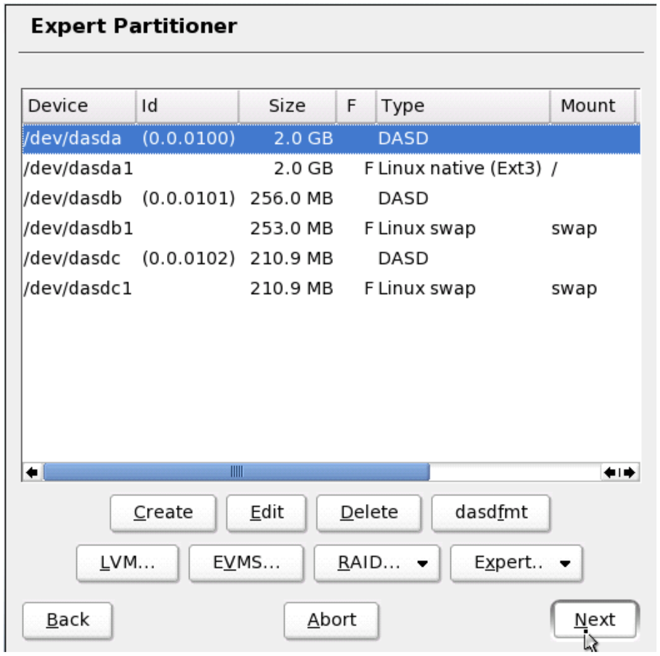
Format
○ Do not format
● Format
File system: Swap

Size
Cylinder size: 0.70 M
Start cylinder: 0
End: (9 or +9M or +3.2GB) 299

Mount Point: swap

Figure 7-4 /dev/dasda1 (root fs) and /dev/dasdc1 (swap) specifications

12. Similarly, highlight **/dev/dasdc1** and click **Edit** as shown in the right side of Figure 7-4. Format this file system as **Swap** and click **OK**. You do not have to format `/dev/dasdb1` because it was properly formatted as a Linux swap space by the **`SWAPGEN EXEC`**, and thus should be recognized as a swap space.
13. Back in the Expert Partitioner window, you should see something similar to Figure 7-5. Click **Next** to continue the installation.



Device	Id	Size	F	Type	Mount
/dev/dasda	(0.0.0100)	2.0 GB		DASD	
/dev/dasda1		2.0 GB	F	Linux native (Ext3)	/
/dev/dasdb	(0.0.0101)	256.0 MB		DASD	
/dev/dasdb1		253.0 MB	F	Linux swap	swap
/dev/dasdc	(0.0.0102)	210.9 MB		DASD	
/dev/dasdc1		210.9 MB	F	Linux swap	swap

Figure 7-5 Disk partitioner after customization

7.3.3 Continuing the YaST2 installation

Follow these steps to continue the YaST2 installation

1. Return to the Installation Settings window, select **Software**.
2. The Software Selection window opens. Select **Minimum graphical system (without KDE)** and click **Accept**.
3. Back in the Installation Settings window, click **Timezone**.
4. In the Clock and Time Zone Configuration window, select your time zone and click **Accept**.
5. Back in the Installation Settings window, you might have to scroll down to click **Language**.
6. In the Language Selection window, select your language and click **Accept**.
7. You are now ready to begin laying down the RPMs onto your root file system. In the Installation Settings window, click **Accept**.
8. A warning window opens to see if you really want to install Linux, click **Yes, install**.

A minimal SLES9 system is installed onto DASD. This should take about five to thirty minutes, depending on network speed and the performance of your NFS server.

If you are using a browser as a VNC client, upon completion, the browser responds `Network error - remote side closed connection`. This is expected because the in-memory Linux system has shutdown.

7.3.4 Booting your new Linux system from disk

After the first part of installation completes, your Linux system shuts down and you return to your z/VM 3270 session. `IPL` the newly installed system from disk to continue the installation. Issue the following command in Figure 7-15 and your new kernel should boot from disk.

Example 7-15 Using ipl 100

```
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: Booting default (ipl)...
Linux version 2.6.5-7.97-s390 (geeko@buildhost) (gcc version 3.3.3 (SuSE Linux))
 #1 SMP Fri Jul 2 14:21:59 UTC 2004
We are running under VM (31 bit mode)
This machine has an IEEE fpu
...
*** You can connect to 129.40.178.127, display :1 now with vncviewer
*** Or use a Java capable browser on http://129.40.178.127:5801/
***
(When YaST2 is finished, close your VNC viewer and return to this window.)
...
```

7.3.5 Completing YaST2 installation

Return to the same browser you used for the first part of installation, and do one of the following:

- Click **Login Again** or click the browser's **refresh** button until you see a VNC login screen.
- Open a new browser window.

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 the one used to first bring up YaST2.
2. The first window you see is for setting the root password. Enter your desired root password *twice* and click **Next**. Do not forget this password!
3. In the Network Configuration window all the values should be correct, so just click **Next**.
4. In the Test Internet Connection window, select **No, skip this test** and click **Next**.
5. The next window is Service Configuration. Select the **Skip configuration** radio button and click **Next**.
6. In the User Authentication Method window accept the default of **Local (/etc/passwd)** and click **Next**.
7. The next window is Add a new local user. You can choose to add a local user so as to have a user other than root on all cloned systems. When you are finished, click **Next**.
8. In the Writing the system configuration window, the `SuSEconfig` tool writes all your settings to disk.
9. The next window is Release Notes. After reviewing the release notes, click **Next** and then **Finish.** You are finished installing Linux!

Return to the 3270 session. You might have to clear the screen a few times. Your new Linux will finish its configuration, but you can again disconnect with the command:

```
==> #cp disc
```

From this point forward, it is recommended that you access your Linux systems with SSH. If you have a Windows desktop, but do not have an SSH client configured, see 3.1, "PuTTY: A free SSH client for Windows" on page 18.

Start an SSH session into the master image as root.

7.4 Configuring the master image

Now you want to customize the master image as much as possible before cloning. The following high level steps are recommended, though you can add or omit some steps:

- 7.4.1, "Applying service if necessary"
- 7.4.2, "Copying files used on the master image" on page 101
- 7.4.3, "Removing unnecessary RPMs" on page 102
- 7.4.5, "Turning off unneeded services" on page 103
- 7.4.6, "Configuring rsyncd" on page 103
- 7.4.7, "Configuring SITAR" on page 104
- 7.4.8, "Setting the software clock accurately" on page 105
- 7.4.9, "Setting system to halt on SIGNAL SHUTDOWN" on page 106
- 7.4.10, "Turning off the hz_timer" on page 106
- 7.4.11, "Hardening the system with Bastille Linux" on page 106
- 7.4.12, "Configuring the VNC server" on page 107
- 7.4.14, "Verifying the changes" on page 108

7.4.1 Applying service if necessary

There are two methods of applying service on SuSE SLES:

- An update CD (Service Pack)
- A YaST Online Update (Y.O.U.)

SLES service packs can be applied to a running system, or they can be built into the install tree. The latter is addressed in 6.2, "Setting up an SLES9 install tree" on page 76. Applying a service pack to a running system is not addressed in this book, but it is documented in the IBM Redbook *IBM Lotus® Domino® 6.5 for Linux on zSeries Implementation*, SG24-7021, on the Web:

`http://www.redbooks.ibm.com/abstracts/sg247021.html`

The remainder of this section describes applying service with YaST Online Update (YOU).

If you have a SuSE Maintenance Web account, then you can use it to retrieve the latest patches for SLES9. Because many of these patches contain security and bug fixes, it is recommended that you apply the patches for the master image so that it is up-to-date. Subsequently all the servers you clone after the master image will also be up-to-date.

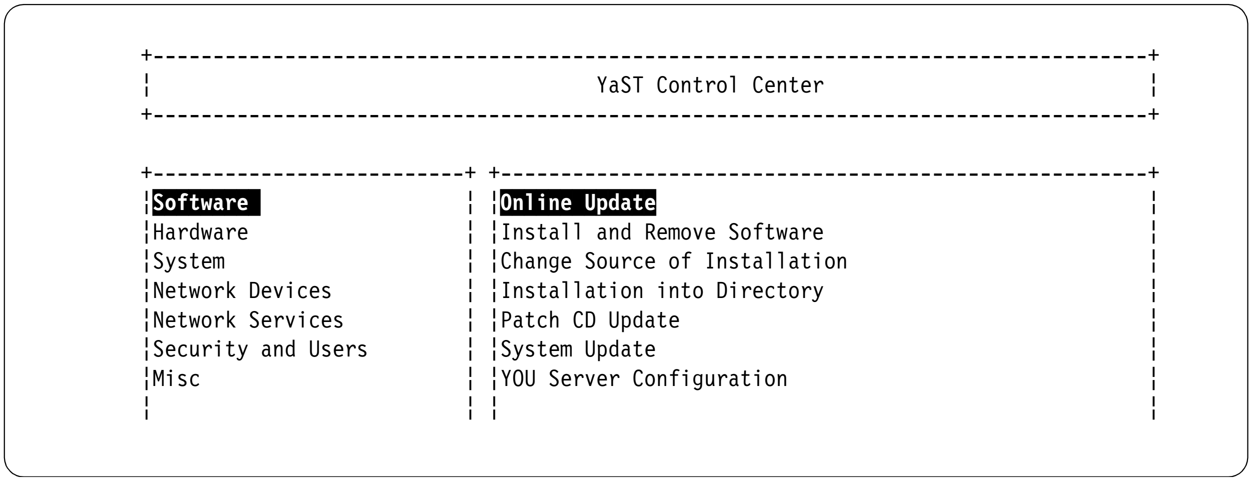
1. Start an SSH session as root to the master Linux server.
2. The Linux installation used **yast2** which is a graphical tool. However, regularly using graphical applications can require many more CPU cycles than their curses applications. For this reason, the curses-based **yast**, is recommended over the graphical **yast2**.

   Bring up **yast** so you can apply the latest patches:

   ```
   # yast
   ```

3. The YaST Control Center opens (Figure 7-6). From the curses menu, **Software** should be selected on the left side. Use the **Tab** key to move to the right side and select **Online Update**. Press Enter.

Figure 7-6 YaST Control Center

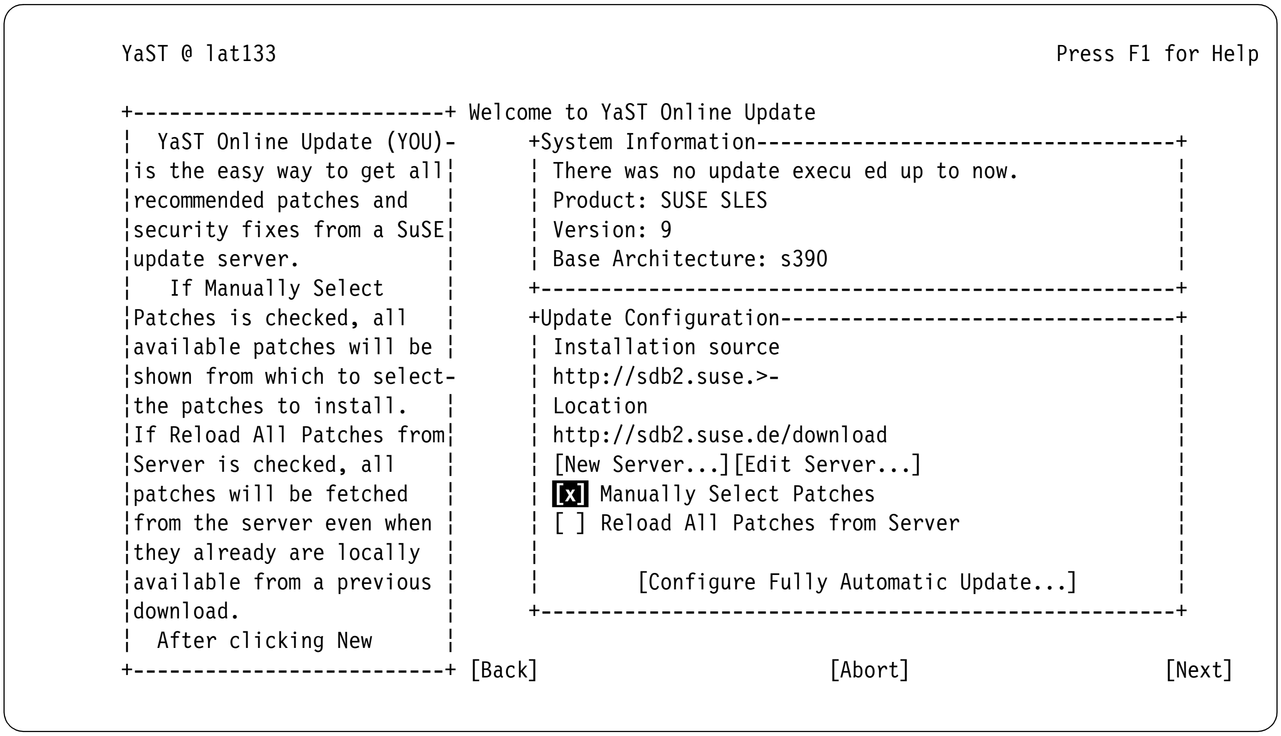


```
+-----------------------------------------------------------------------------------+
¦                                 YaST Control Center                               ¦
+-----------------------------------------------------------------------------------+

+---------------------------+ +-----------------------------------------------------+
¦Software                   ¦ ¦Online Update                                        ¦
¦Hardware                   ¦ ¦Install and Remove Software                          ¦
¦System                     ¦ ¦Change Source of Installation                        ¦
¦Network Devices            ¦ ¦Installation into Directory                          ¦
¦Network Services           ¦ ¦Patch CD Update                                      ¦
¦Security and Users         ¦ ¦System Update                                        ¦
¦Misc                       ¦ ¦YOU Server Configuration                             ¦
¦                           ¦ ¦                                                     ¦
```

4. The Online Update panel displays the default YaST Online Update server information. You need access to the Internet to resolve the default server of **sdb2.suse.de** successfully. Leave the default settings, use the Tab key to select **Next,** and press Enter.

Note: If you do not have a correct DNS setup, YaST cannot reach the update server and never reaches the Online Update panel. In this situation, it is recommended that you fix your DNS settings. If you do not have a DNS, edit the file `/etc/sysconfig/onlineupdate` and change the setting YAST2_LOADFTPSERVER to **no**. Then, YaST will not try to load the update server before getting to the Online Update panel. When you see the Online Update panel, enter the IP address of the server.

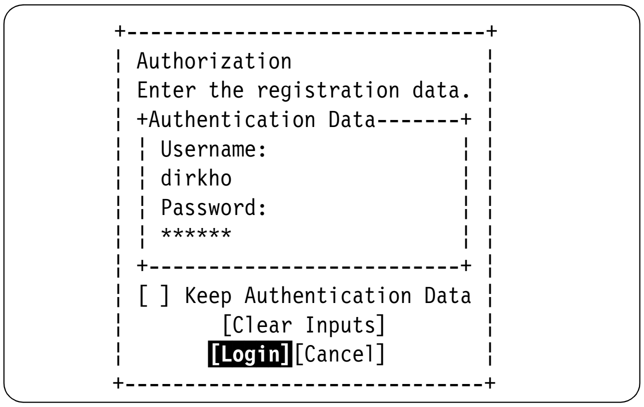
Figure 7-7 YaST Online Update panel



```
YaST @ lat133                                                                  Press F1 for Help

+--------------------------+ Welcome to YaST Online Update
¦  YaST Online Update (YOU)-      +System Information-----------------------------------+
¦is the easy way to get all¦      ¦ There was no update execu ed up to now.             ¦
¦recommended patches and   ¦      ¦ Product: SUSE SLES                                  ¦
¦security fixes from a SuSE¦      ¦ Version: 9                                          ¦
¦update server.            ¦      ¦ Base Architecture: s390                             ¦
¦   If Manually Select     ¦      +-----------------------------------------------------+
¦Patches is checked, all   ¦      +Update Configuration---------------------------------+
¦available patches will be ¦      ¦ Installation source                                 ¦
¦shown from which to select-      ¦ http://sdb2.suse.>-                                 ¦
¦the patches to install.   ¦      ¦ Location                                            ¦
¦If Reload All Patches from¦      ¦ http://sdb2.suse.de/download                        ¦
¦Server is checked, all    ¦      ¦ [New Server...][Edit Server...]                     ¦
¦patches will be fetched   ¦      ¦ [x] Manually Select Patches                         ¦
¦from the server even when ¦      ¦ [ ] Reload All Patches from Server                  ¦
¦they already are locally  ¦      ¦                                                     ¦
¦available from a previous ¦      ¦       [Configure Fully Automatic Update...]         ¦
¦download.                 ¦      +-----------------------------------------------------+
¦  After clicking New      ¦
+--------------------------+ [Back]                      [Abort]                    [Next]
```

5. You are prompted for your maintenance Web user ID and password (Figure 7-8). Enter them and use the Tab key to move to **Login** and press **Enter**:

Figure 7-8 Longin



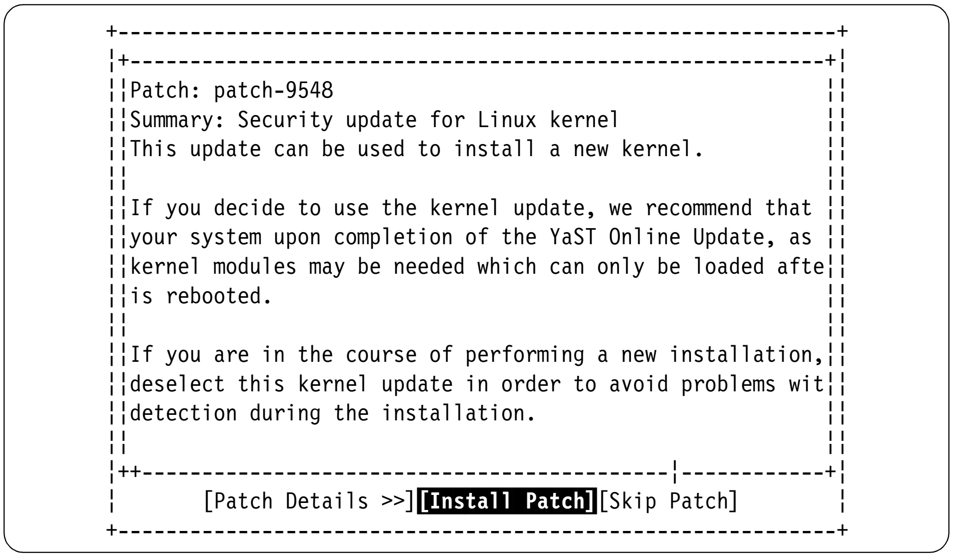
```
+------------------------------+
¦ Authorization                ¦
¦ Enter the registration data. ¦
¦ +Authentication Data-------+ ¦
¦ ¦ Username:                ¦ ¦
¦ ¦ dirkho                   ¦ ¦
¦ ¦ Password:                ¦ ¦
¦ ¦ ******                   ¦ ¦
¦ +--------------------------+ ¦
¦ [ ] Keep Authentication Data ¦
¦        [Clear Inputs]        ¦
¦       [Login][Cancel]        ¦
+------------------------------+
```

6. A progress is displayed as updates applicable to your system are retrieved. When the updates complete, you should see a list of patches available for your system. Notice that, by default, all the patches have a **+** mark on the left-most column, this is to indicate that they are all to be applied. If you do not want to apply certain updates, simply highlight

them one by one and enter **-** (a minus sign) so that they are deselected. After you have done that, Tab to **OK** and press **Enter**.

7. Another panel opens, providing progress information for applying the updates. While it is rare, you might encounter a window prompting you for action, especially if it is a kernel patch that would require a reboot or running **`zipl`**. Figure 7-9 is one such window. Because you already installed Linux, select **Install Patch** anyway:

Figure 7-9 Progress update panel: applying a kernal patch



```
+----------------------------------------------------------+
¦+--------------------------------------------------------+¦
¦¦Patch: patch-9548                                       ¦¦
¦¦Summary: Security update for Linux kernel               ¦¦
¦¦This update can be used to install a new kernel.        ¦¦
¦¦                                                        ¦¦
¦¦If you decide to use the kernel update, we recommend that ¦¦
¦¦your system upon completion of the YaST Online Update, as ¦¦
¦¦kernel modules may be needed which can only be loaded afte¦¦
¦¦is rebooted.                                            ¦¦
¦¦                                                        ¦¦
¦¦If you are in the course of performing a new installation,¦¦
¦¦deselect this kernel update in order to avoid problems wit¦¦
¦¦detection during the installation.                      ¦¦
¦¦                                                        ¦¦
¦++-------------------------------------------¦-----------+¦
¦       [Patch Details >>][Install Patch][Skip Patch]       ¦
+----------------------------------------------------------+
```

8. When you are almost finished, you might receive a message stating the following:

   ```
   After the kernel is updated, please run zipl
   ```

   If you do, tab to **OK** and press Enter**.**

9. Tab to the **Finish** button and press Enter. The **`SuSEconfig`** tool runs and writes the changes to disk. On the main menu, tab to **Quit** to end **`yast`**.

10. If you were asked to run **`zipl,`** do so now and then **`reboot`** (Example 7-16).

Example 7-16 Running zip1 after kernal installation

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap '/boot/zipl/bootmap'
Adding IPL section 'ipl' (default)
Preparing boot device: dasda (0103).
Done.
# reboot
```

After you type **`reboot,`** you will lose your SSH session. Wait a couple of minutes for the system to come back up to **Start another SSH session**.

7.4.2 Copying files used on the master image

There are files associated with this book that should be copied from the NFS server to the controller. For convenience, they are in the correct hierarchy on the NFS server so all can be copied at once. These files are listed here:

- `/etc/Bastille/config`
- `/etc/cron.daily/set-clock`
- `/etc/cron.daily/run-sitar`

- `/etc/ntp.conf.template`
- `/etc/rsyncd.conf`

Each of these files is addressed in the sections that follow. Use the secure copy **`scp -rp`** command to recursively copy these files from your NFS server to the root of the file system. You will need to provide the root password of the NFS server.

Example 7-17 Copying files with scp -rp

```
# scp -rp <129.40.46.206>:/nfs/virt-cookbook/linux-master/* /
The authenticity of host '129.40.46.206 (129.40.46.206)' can't be established.
RSA key fingerprint is cc:5e:29:0e:9c:c3:8b:3f:1b:6a:98:46:fd:df:e2:dc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '129.40.46.206' (RSA) to the list of known hosts.
root@129.40.46.206's password:
ntp.conf.template                             100% 2057     2.0KB/s   00:00
rsyncd.conf                                   100%  247     0.2KB/s   00:00
config                                        100% 2238     2.2KB/s   00:00
set-clock                                     100%   49     0.1KB/s   00:00
run-sitar                                     100%   58     0.1KB/s   00:00
```

7.4.3 Removing unnecessary RPMs

For the master image, it is desirable to have as lean a Linux image as possible from which to build. It is recommended that the following RPMs be removed. You can choose to add to or omit some from this list.

eject	For CDs and tapes
ethtool	For examining and tuning Ethernet-based network interfaces
fribidi	A Free Implementation of the Unicode BiDi algorithm
fvwm2	A window manager
hfsutils	Is for Apple MacIntosh computers
initviocons	For IBM iSeries™ machines
ntfsprogs	Utilities for the Windows NTFS filesystem
pmtools	A set of tools to display your BIOS ACPI tables
providers	A list of Internet service providers
unclutter	Hide the mouse cursor after inactivity
usbutils	Tools and libraries for USB devices
wol	Wake On Lan client
xfsprogs	Utilities for managing the XFS file system
zsh	A shell designed for interactive use

Start an SSH session to your new Linux server if you do not already have one. Issue the following **`rpm`** command:

```
# rpm -e eject fvwm2 hfsutils initviocons ntfsprogs pmtools providers unclutter \
usbutils wol xfsprogs zsh
```

7.4.4 Adding additional RPMs

Add additional RPMs using the **`yast`** command with the **`-i`** option. YaST will conveniently install the packages specified and automatically resolve their dependencies. You can choose the RPMs you want to add.

- The findutils-locate package will be used to enable the **`locate`** and **`updatedb`** commands.
- The bastille package is a Bastille Linux system hardening tool used to *tighten* security on your system.

- The xntp package is the Network Time Protocol daemon is needed for synchronizing the software clock.
- The rsync package will be used to backup each virtual server's `/etc/` subdirectory.
- The sitar package will be used to produce a report on the server's configuration.
- The wget package will be used to get other packages off the Internet.
- The openmotif package will be used by the VNC server as a usable window manager.

You can choose to add other RPMs, or to omit some of the above. To add the RPMs described above, use the **`yast -i`** command:

```
# yast -i findutils-locate bastille xntp rsync sitar wget openmotif
```

You should see **`yast`** curses screens go by as the software is added.

7.4.5 Turning off unneeded services

There are a number of services which are started in a SLES9 minimum system. In order to keep the master image as lean as possible, some of these can be turned off:

nfs	The Network File System
slpd	The Service Location Protocol daemon

Turn off the following services by using the **`chkconfig`** command:

- `# chkconfig nfs off`
- `# chkconfig slpd off`

You can 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 command in Example 7-18.

Example 7-18 Using chkconfig -l grep3:on

```
# chkconfig -l | grep 3:on
coldplug                  0:off  1:on   2:on   3:on   4:off  5:on   6:off
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
fbset                     0:off  1:on   2:on   3:on   4:off  5:on   6:off
hwscan                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
kbd                       0:off  1:on   2:on   3:on   4:off  5:on   6:off
network                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfsboot                   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
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
```

7.4.6 Configuring rsyncd

It is important for Linux configuration files to be backed up. There are many ways to accomplish this and one of them is **`rsync`**. Rather than copying entire files as with the **`scp -rp`** command, **`rsync`** just copies the differences or deltas of changed files. For this reason it is an efficient tool for backups.

Use the **`chkconfig`** command to set rsync to start up:

```
# chkconfig rsync on
```

There are two files that can be configured: the main configuration file `/etc/rsyncd.conf` and a file to allow only the controller to connect, `/etc/rsyncd.secrets`. The file `/etc/rsyncd.conf` should have been copied when you copied all files from the NFS server. It will export the `/etc/` directory to the controller.

Example 7-19 Using /etc/rsyncd.conf

```
# cd /etc
# cat rsyncd.conf
gid = users
read only = yes
transfer logging = true
log format = %h %o %f %l %b
hosts allow = /etc/trusted.hosts
max connections = 1
log file = /var/log/rsync.log
timeout = 300

[etc]
        path = /etc
        read only = yes
        list = yes
```

The file `/etc/trusted.hosts` needs a single entry which is the TCP/IP address of the controller. This will allow only the controller to get an **rsync** session to the virtual servers.

Create this file with the **vi** editor:

```
# vi trusted.hosts    // add one line - the TCP/IP address of the controller
<129.40.178.127>
```

7.4.7 Configuring SITAR

SITAR is a tool that creates documentation describing your system. It is an acronym for System InformaTion At Runtime. In addition to backing up the `/etc/` configuration files, a document describing your system might be helpful. If SITAR is run once a day and the output is put in a file in `/etc/`, then you will have useful information about your system backed up.

An HTML file, written to `/etc/sitar.html`, describing your system configuration can be created with the following command. And then you can view your system information in `/etc/sitar.html`.

*Example 7-20 Creating an *.html file*

```
# sitar --format=html --outfile /etc/sitar.html
Disk /dev/dasda doesn't contain a valid partition table
Disk /dev/dasdb doesn't contain a valid partition table
Disk /dev/dasdc doesn't contain a valid partition table
/usr/bin/find: /etc/lvm/backup/: No such file or directory
```

The errors stating that disks do not contain a valid partition table should not be a problem. You can view the resulting HTML file, `/etc/sitar.html` with a browser.

You should have also copied the **run-sitar** in the directory `/etc/cron.daily/` so that it is run every night.

```
# cat /etc/cron.daily/run-sitar
#!/bin/bash
sitar --format=html --outfile=/etc/sitar.html
```

Because the output is sent to the `/etc/` directory and because that directory is backed up nightly with rsync (if all the customization steps are completed successfully), you will have a copy of the **`sitar`** output backed up once a day.

7.4.8 Setting the software clock accurately

It is important to have the Linux software clock set properly. Commonly, this is accomplished by running the **`xntpd`** daemon against some accurate time sources. However, this is expensive in terms of CPU costs. A compromise between an extremely accurate clock (running **`xntpd`**) and using fewer CPU cycles (not running it) is to reset the clock once a day. This can be done with a script that runs the **`ntpd -q`** command from the `/etc/cron.daily/` directory. NTP will be configured so the controller will be the external clock. In 7.6.8, “Turning on the NTP server” on page 117 the controller is configured to point to external clocks on the Internet.

When you copied the files in 7.4.2, “Copying files used on the master image” on page 101, you should have copied the file `/etc/ntp.conf.template`. It has two lines commented out that point to an `Undisciplined Local Clock`. It adds one line pointing to the external time reference (Note: for better accuracy, you should have a second external clock), however the IP address is `n.n.n.n`. Back up the original `ntp.conf` file and change the value `n.n.n.n` to be the IP address of the controller as in Example 7-21.

Example 7-21 Backing up /etc/ntp.conf.template with an external clock IP address

```
# cd /etc
# cp ntp.conf ntp.conf.orig
# mv ntp.conf.template ntp.conf
# 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 <129.40.178.127>
...
```

You should have also copied the file `/etc/cron.daily/set-clock`. This is a small script that will run once a day to reset the Linux software clock via the **`ntpd -q`** command:

Example 7-22 Runing /etc/cron.daily/set-clock

```
# cd /etc/cron.daily
# cat set-clock
#!/bin/bash
# Adjust the clock
/usr/sbin/ntpd -q
```

Now you should have the configuration file `/etc/ntp.conf` pointing to the controller, and a **`set-clock`** script in the directory `/etc/cron.daily/` that will run once a day to set the software clock accurately.

7.4.9 Setting system to halt on SIGNAL SHUTDOWN

By default, SLES9 SP3 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, you want your system to halt (shutdown).

Change this setting by changing **shutdown -r** to **shutdown -h** in the `/etc/inittab` file:

Example 7-23 Halt on shutdown

```
# cd /etc
# 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 in effect when the system is rebooted.

7.4.10 Turning off the hz_timer

By default, the Linux kernel wakes up 100 times per second to see if there is any work to be done. While this is fine for a PC running a single copy of Linux, it can consume CPU cycles as the number of virtual servers goes up. A rule of thumb on zSeries is to turn off this timer unless the server has a heavy, constant workload.

Turning off the `hz_timer` can be accomplished by adding a **sysctl** command to the file `/etc/init.d/boot.local` which is run each time the virtual server is booted (Example 7-24).

Example 7-24 Turning off the hz_timer

```
# vi /etc/init.d/boot.local    // add one line at the bottom
#! /bin/sh
...#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
sysctl -w kernel.hz_timer=0
```

Before shutting down, note that the `hz_timer` is on:

```
# cat /proc/sys/kernel/hz_timer
1
```

When you reboot the system, the hz_timer, or timer pop, should be off.

7.4.11 Hardening the system with Bastille Linux

Bastille Linux is a great tool to harden a Linux system. It tightens access control, stops unnecessary services from running, and tightens root access. The file `/etc/Bastille/config` should have been copied from the NFS server. You can look into this file to see the hardening settings.

The **bastille** flag **-b** specifies using the configuration file. You will have to type `accept` to agree with the user agreement (Example 7-25).

Example 7-25 Invoking Bastille Linux to harden the system

```
# bastille -b
Copyright (C) 1999-2002 Jay Beale
Copyright (C) 1999-2001 Peter Watkins
Copyright (C) 2000 Paul L. Allen
Copyright (C) 2001-2003 Hewlett Packard Company
...
You must accept the terms of this disclaimer to use
Bastille.  Type "accept" (without quotes) within 5
minutes to accept the terms of the above disclaimer
> accept
...
########################################################
Errors have occurred in the configuration.
Please view the following file for more details:
        /var/log/Bastille/error-log
########################################################
```

The last message that errors have occurred should be benign. The code is looking for the file `/etc/grub.conf` which is not used on the s390[x] architecture. The master image should now be hardened.

7.4.12 Configuring the VNC server

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 with the **`vncserver`** command. The `openmotif` package allows for the motif window manager (**`mwm`**) that is more usable than the default Tiny Window Manager (**`twm`**) that VNC uses by default.

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 (Example 7-26)

Example 7-26 Setting your VNC server password

```
# 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

New 'X' desktop is pbc4553:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/pbc4553:1.log
```

It is recommended, but not required, to change the window manager from the Tiny Window Manger (**`twm`**) to openmotif (**`mwm`**). To do this, first stop the VNC server using the **`-kill :1`** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 2621C
```

Change the window manger from **`twm`** to **`mwm`** in the file `/root/.vnc/xstartup` (Example 7-27).

Example 7-27 Changing the window manager to mwm

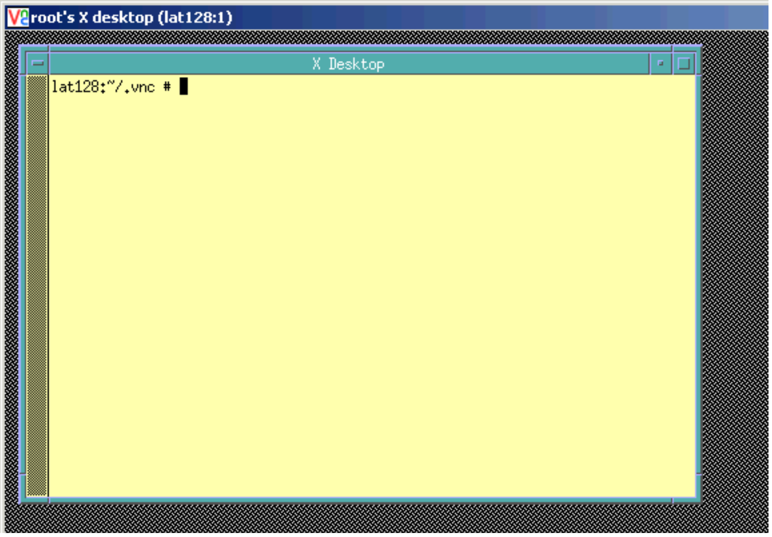
```
# cd /root/.vnc/
# vi xstartup
#!/bin/sh
```

```
xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

Remove the `passwd` file so the cloned system does not have the same password as you just entered.

```
# rm passwd
```

When a system is cloned, the password will be prompted for the first time that the VNC server is initialized. A sample session is shown in Figure 7-10.



```
lat128:~/.vnc #
```

Figure 7-10 VNC client session to the VNC server

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`.

7.4.13 Rebooting the system

It is recommended that you **`reboot`** to test your changes:

```
# reboot

Broadcast message from root (pts/0) (Thu Nov 18 15:50:57 2004):

The system is going down for reboot NOW!
```

7.4.14 Verifying the changes

You are now finished with customizing the master Linux image. When the system comes back up, verify the changes that you made.

1. SSH back into the master image and check a few settings.
2. Use the **`df`** command to display your file systems. Note that out of a 2GB root file system you are using about a third of it (your output might differ):

```
# df -h
```

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.1G  629M  1.4G  32% /
tmpfs                 124M  4.0K  124M   1% /dev/shm
```

3. Confirm that both of your swap spaces are operational:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdb1                             partition       259956  0       42
/dev/dasdc1                             partition       215896  0       42
```

4. Verify that the `hz_timer` is off:

```
# cat /proc/sys/kernel/hz_timer
0
```

5. Shutdown your master image from the SSH session. Next you will install the controller image.

```
# shutdown -h now
```

Congratulations! You have now successfully installed the master image. This image will normally be shut down or *quiesced*. It is now time to install the controller Linux image which will normally be running.

7.5 Installing the controller

In this section you install the controller Linux image.

1. **Logon to SLES9X** or resume your 3270 session. You should see messages that result from the master image being shut down.
2. At this point, you could IPL CMS run the **`SLES9X EXEC`** to purge the reader, punch the bootstrap files, and IPL from the reader. However, you would be purging and punching the same three files. You can save some time by simply IPLing from the reader at virtual address `00C` with the **`#CP`** prefix. Think of it as booting your PC from an old-fashioned diskette.

Example 7-28   IPLing the reader

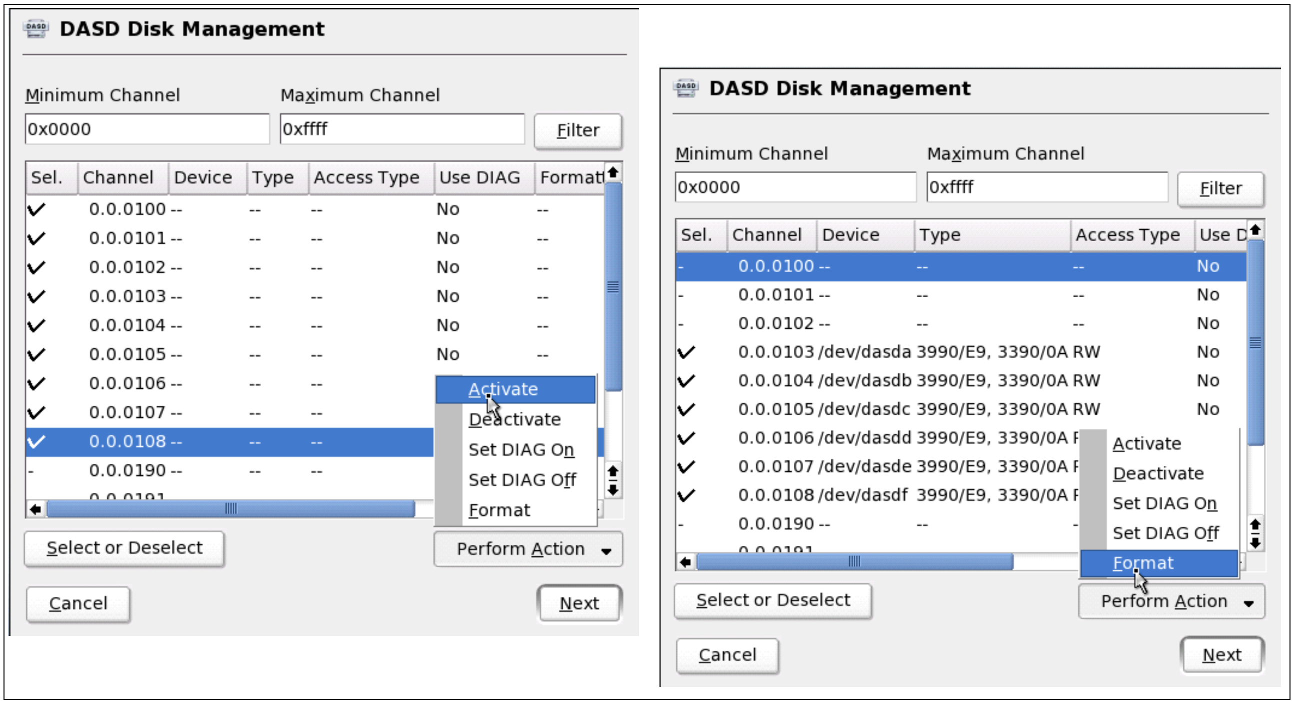
```
...
Power down.
01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop fro
 CPU 00.
00: HCPGSP2630I The virtual machine is placed in CP mode due to a SIGP stop and
store status from CPU 00.

==> #cp ipl 00c
00: 0000003 FILES CHANGED
Linux version 2.6.5-7.97-s390 (geeko@buildhost) (gcc version 3.3.3 (SuSE Linux))
 #1 SMP Fri Jul 2 14:21:59 UTC 2004
We are running under VM (31 bit mode)
...
```

3. Install Linux again as described in 7.3.2, “Beginning YaST installation” on page 93 with the following differences.
   a. When you get to the DASD Disk Management window, the settings are a bit different.
      i. Select all nine read/write disks (**100-108**)
      ii. Click **Activate** as shown on the left side of Figure 7-11 on page 110. The DASD will be activated quickly.

iii. Deselect **100**, **101** and **102** so that **103-108** are selected.

iv. Click **Format** as shown on the right side of the figure. **Do not format 100** because this will undo all the work you just did on the master image. The DASD will be formatted in parallel and will take a number of minutes.

v. Click **Next** when the formatting is complete.



DASD Disk Management

Minimum Channel: 0x0000 Maximum Channel: 0xffff

Sel.	Channel	Device	Type	Access Type	Use DIAG	Format
✓	0.0.0100	--	--	--	No	--
✓	0.0.0101	--	--	--	No	--
✓	0.0.0102	--	--	--	No	--
✓	0.0.0103	--	--	--	No	--
✓	0.0.0104	--	--	--	No	--
✓	0.0.0105	--	--	--	No	--
✓	0.0.0106	--	--	--		
✓	0.0.0107	--	--	--		
✓	0.0.0108	--	--	--		
-	0.0.0190	--	--	--		

DASD Disk Management

Minimum Channel: 0x0000 Maximum Channel: 0xffff

Sel.	Channel	Device	Type	Access Type	Use D
-	0.0.0100	--	--	--	No
-	0.0.0101	--	--	--	No
-	0.0.0102	--	--	--	No
✓	0.0.0103	/dev/dasda	3990/E9, 3390/0A	RW	No
✓	0.0.0104	/dev/dasdb	3990/E9, 3390/0A	RW	No
✓	0.0.0105	/dev/dasdc	3990/E9, 3390/0A	RW	No
✓	0.0.0106	/dev/dasdd	3990/E9, 3390/0A	F	
✓	0.0.0107	/dev/dasde	3990/E9, 3390/0A	F	
✓	0.0.0108	/dev/dasdf	3990/E9, 3390/0A	F	
-	0.0.0190	--	--	-	

Figure 7-11 Activating and formatting DASD on the controller

b. A window opens asking for the type of installation. Select **New installation** and click **OK**.

c. Click **Partitioning** and the Expert Partioner window opens. It is recommended that you format the DASD with the information in Table 7-1.

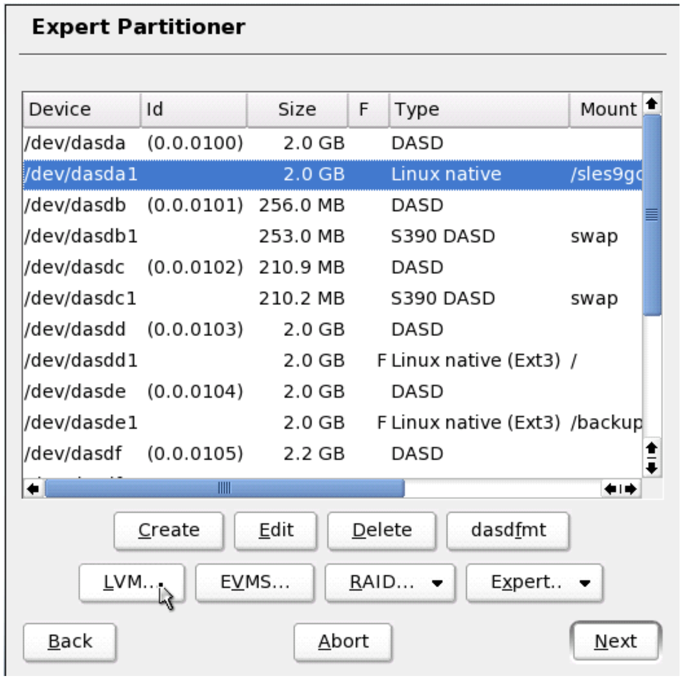
Table 7-1 Controller disk partitioning

minidisk	Linux device	Format	Mount Point	Notes
100	`/dev/dasda1`	No	`/sles9master`	master image - **don't format!**
101	`/dev/dasdb1`	No	swap	Should be detected as a swap space
102	`/dev/dasdc1`	No	swap	Should be detected as a swap space
103	`/dev/dasdd1`	As ext3	/	Controller root file system
104	`/dev/dasde1`	As ext3	`/backup`	For rsync backup

i. Select a mount point of `/sles9master` for **`/dev/dasda1`** (**Note**: it is important that the directory is named `/sles9master/` as it is hard coded into the **`clone.sh`** script)

ii. You should be able to skip `/dev/dasdb1` and `/dev/dasdc1` as they should be recognized as swap spaces.

iii. Format and select mount points for **`/dev/dasdd1`** and **`/dev/dasde1`** as in Table 7-1. The devices should now look similar to Figure 7-12.



Expert Partitioner

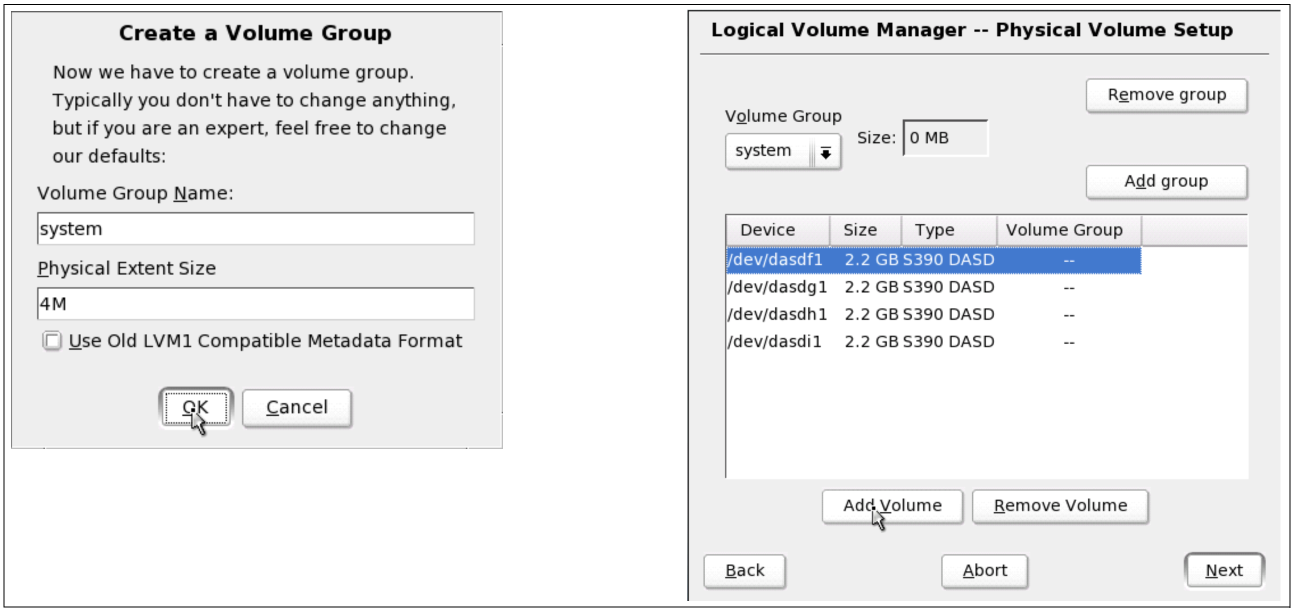
Device	Id	Size	F	Type	Mount
/dev/dasda	(0.0.0100)	2.0 GB		DASD	
/dev/dasda1		2.0 GB		Linux native	/sles9gc
/dev/dasdb	(0.0.0101)	256.0 MB		DASD	
/dev/dasdb1		253.0 MB		S390 DASD	swap
/dev/dasdc	(0.0.0102)	210.9 MB		DASD	
/dev/dasdc1		210.2 MB		S390 DASD	swap
/dev/dasdd	(0.0.0103)	2.0 GB		DASD	
/dev/dasdd1		2.0 GB	F	Linux native (Ext3)	/
/dev/dasde	(0.0.0104)	2.0 GB		DASD	
/dev/dasde1		2.0 GB	F	Linux native (Ext3)	/backup
/dev/dasdf	(0.0.0105)	2.2 GB		DASD	

Figure 7-12 Expert Partitioner before LVM

iv. For `/dev/dasdf1-/dev/dasdi1` (four devices) you will create a logical volume. Click **LVM** in the Expert Partitioning window as is shown in Figure 7-12.

d. The window Create a Volume Group will appear as is shown on the left side of Figure 7-13 on page 112. Click **OK** to accept a volume group name of **system** and a **4MB** physical extent size.

e. The window Logical Volume Manager -- Physical Volume Setup opens, as shown on the right side of Figure 7-13 on page 112. The device `/dev/dasdf1` should be selected. Click **Add Volume**. Select each of the other three volumes and click **Add Volume**. The volume group should now have 9.1GB of space. Click **Next**.



Create a Volume Group

Now we have to create a volume group. Typically you don't have to change anything, but if you are an expert, feel free to change our defaults:

Volume Group Name: system

Physical Extent Size: 4M

☐ Use Old LVM1 Compatible Metadata Format

Logical Volume Manager -- Physical Volume Setup

Volume Group: system

Size: 0 MB

Device	Size	Type	Volume Group
/dev/dasdf1	2.2 GB	S390 DASD	--
/dev/dasdg1	2.2 GB	S390 DASD	--
/dev/dasdh1	2.2 GB	S390 DASD	--
/dev/dasdi1	2.2 GB	S390 DASD	--

Figure 7-13 Creating a volume group

f. The window Logical Volume Manager -- Logical Volumes opnes as shown in the left side of Figure 7-14 on page 113. This shows the 9.1GB of free space with which logical volumes can be created. Click **Add** to create a logical volume.

g. The window Create a Logical Volume opnes as shown in the right side of Figure 7-14.

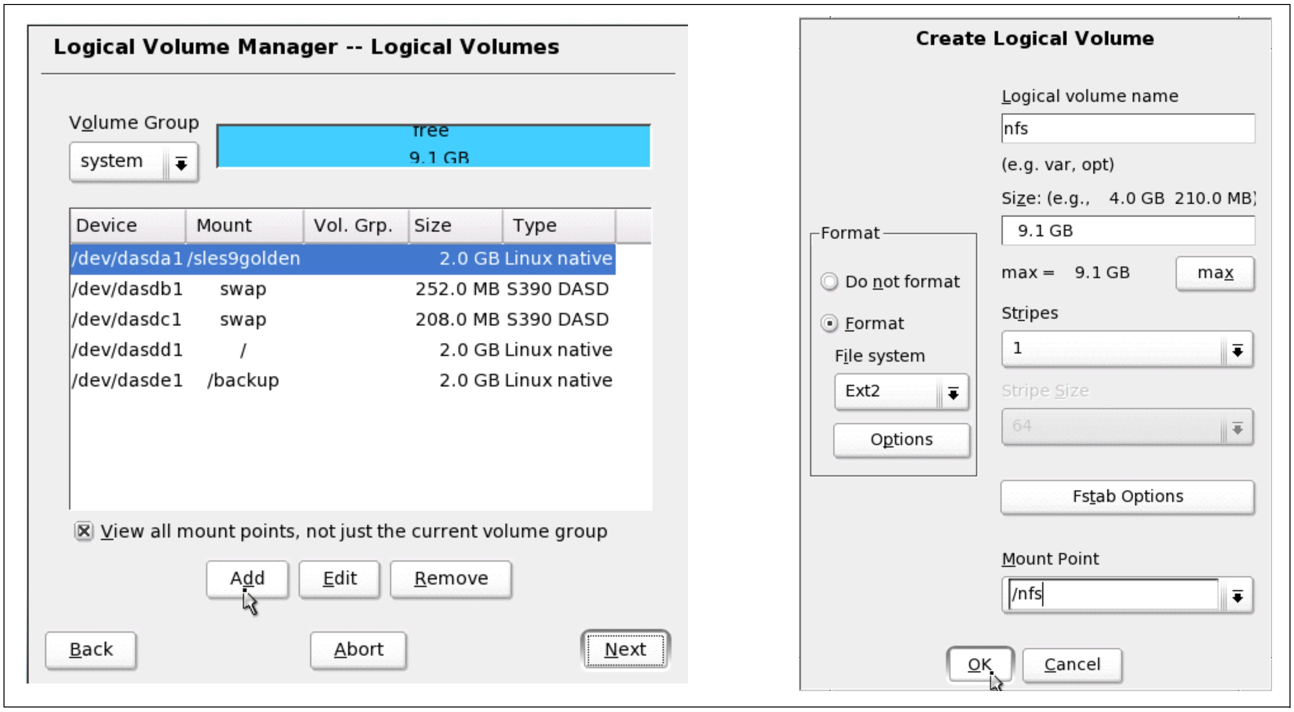
   i. Select a logical volume name. In this example, **nfs** is chosen.

   ii. Click the button **max** to dedicate all of the space to this logical volume.

   iii. **Format** the file system as **ext2** (a journalled file system such as ext3 is probably not necessary as this file system will seldom be written to)

   iv. Specify the mount point. In this example the directory `/nfs/` is chosen.

   v. Click **OK** to create the logical volume.



Logical Volume Manager -- Logical Volumes

Volume Group: system

free 9.1 GB

Device	Mount	Vol. Grp.	Size	Type
/dev/dasda1	/sles9golden		2.0 GB	Linux native
/dev/dasdb1	swap		252.0 MB	S390 DASD
/dev/dasdc1	swap		208.0 MB	S390 DASD
/dev/dasdd1	/		2.0 GB	Linux native
/dev/dasde1	/backup		2.0 GB	Linux native

☒ View all mount points, not just the current volume group

Create Logical Volume

Logical volume name: nfs

(e.g. var, opt)

Size: (e.g., 4.0 GB 210.0 MB): 9.1 GB

max = 9.1 GB

Format

○ Do not format

● Format

File system: Ext2

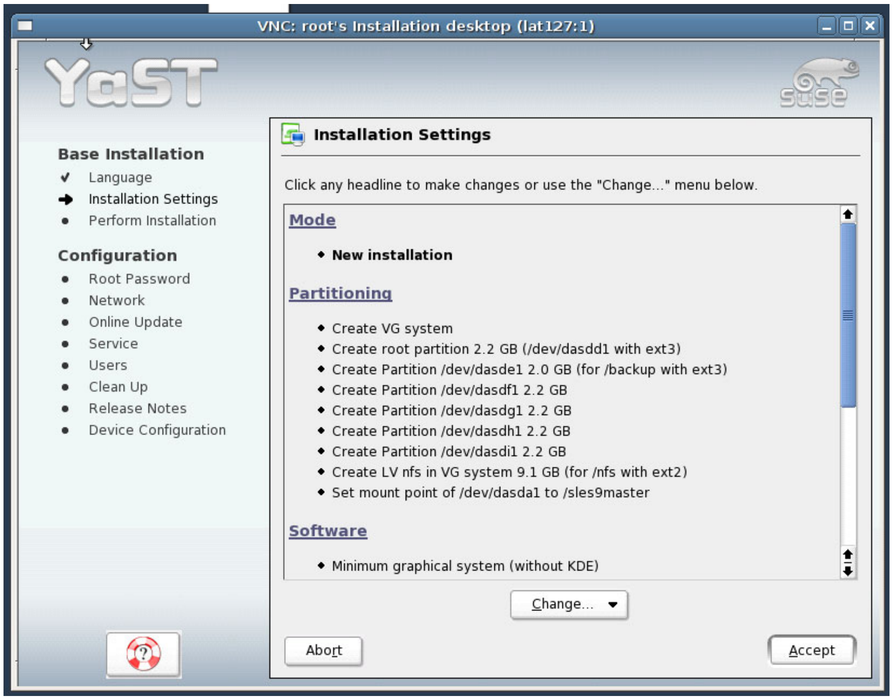
Stripes: 1

Stripe Size: 64

Mount Point: /nfs

Figure 7-14 Creating a logical volume

- Click **Next** in the Logical Volume Manager -- Logical Volumes window.
- Click **Next** in the Expert Partitioner window. When you are finished, the partitions should be similar or identical to those shown in Figure 7-15.



VNC: root's Installation desktop (lat127:1)

Base Installation

- ✓ Language
- → Installation Settings
- Perform Installation

Configuration

- Root Password
- Network
- Online Update
- Service
- Users
- Clean Up
- Release Notes
- Device Configuration

Installation Settings

Click any headline to make changes or use the "Change..." menu below.

Mode

- **New installation**

Partitioning

- Create VG system
- Create root partition 2.2 GB (/dev/dasdd1 with ext3)
- Create Partition /dev/dasde1 2.0 GB (for /backup with ext3)
- Create Partition /dev/dasdf1 2.2 GB
- Create Partition /dev/dasdg1 2.2 GB
- Create Partition /dev/dasdh1 2.2 GB
- Create Partition /dev/dasdi1 2.2 GB
- Create LV nfs in VG system 9.1 GB (for /nfs with ext2)
- Set mount point of /dev/dasda1 to /sles9master

Software

- Minimum graphical system (without KDE)

Figure 7-15 Installation settings - partitions defined

4. Complete the remainder of the installation as you did for the master image described in 7.3.3, “Continuing the YaST2 installation” on page 96, but do not continue with 7.3.4, “Booting your new Linux system from disk” on page 97.
5. After the initial system is shutdown, you must IPL from minidisk 103, not `100`, the master image (Example 7-29).

Example 7-29 IPLing from 103

```
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 103
...
```

6. Complete the installation as described in 7.3.5, “Completing YaST2 installation” on page 97. When your system is completely installed, you can **`DISCONNECT`** from to the 3270 session using the **`#CP`** prefix.

```
==> #cp disc
```

7.5.1 Verifying the installation

Start a new SSH session to the controller. You might see a warning from PuTTY about a “POTENTIAL SECURITY BREACH”. This is expected because a new set of SSH keys were generated for the same IP address the second time you installed. Click **Yes** to begin the session.

Verify some settings with the **`mount`** and **`swapon`** commands. You should see the following file systems mounted and the two swap spaces (Example 7-30).

Example 7-30 Verifying settings

```
# mount
/dev/dasdd1 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
tmpfs on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/dasde1 on /backup type ext3 (rw,acl,user_xattr)
/dev/mapper/system-nfs on /nfs type ext2 (rw,acl,user_xattr)
/dev/dasda1 on /sles9master type ext3 (rw)
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdb1                             partition       259956  0       42
/dev/dasdc1                             partition       215896  0       42
```

7.6 Configuring the controller

Now that your controller is installed, it must be configured. The following steps are involved:

- 7.6.1, “Applying service if necessary” on page 115
- 7.6.2, “Copying files to the controller” on page 115
- 7.6.3, “Removing unnecessary RPMs” on page 116
- 7.6.4, “Adding additional RPMs” on page 116
- 7.6.5, “Installing the cmsfs package” on page 116
- 7.6.6, “Turning off an unneeded service” on page 117
- 7.6.7, “Turning on the NFS server” on page 117

- 7.6.8, “Turning on the NTP server” on page 117
- 7.6.9, “Enabling the vmcp module” on page 119
- 7.6.10, “Setting system to halt on SIGNAL SHUTDOWN” on page 120
- 7.6.11, “Turning off the hz_timer” on page 120
- 7.6.12, “Configure SSH keys” on page 120
- 7.6.13, “Hardening the controller” on page 121
- 7.6.15, “Verifying the changes” on page 121

7.6.1 Applying service if necessary

This is the same step done for the master image, so refer to 7.4.1, “Applying service if necessary” on page 98. If you do apply service, remember to run `mkinitrd`, `zipl` and `reboot` if these steps are necessary.

7.6.2 Copying files to the controller

The following files associated with this book should be copied to the controller. Use the `scp -rp` command to recursively copy these files from your NFS server (Example 7-31).

Example 7-31 Using scp -rp to copy files to the controller

```
# scp -rp <129.40.46.206>:/nfs/virt-cookbook/linux-controller/* /
The authenticity of host '129.40.46.206 (129.40.46.206)' can't be established.
RSA key fingerprint is cc:5e:29:0e:9c:c3:8b:3f:1b:6a:98:46:fd:df:e2:dc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '129.40.46.206' (RSA) to the list of known hosts.
root@129.40.46.206's password:
ntp.conf.template                                      100% 2082     2.0KB/s   00:00
config                                                 100% 2238     2.2KB/s   00:00
TCPMAINT.FTPcommands                                   100%   98     0.1KB/s   00:00
LNXMAINT.FTPcommands                                   100%  149     0.2KB/s   00:00
backup_linux                                           100% 1528     1.5KB/s   00:00
backup_vm                                              100% 1563     1.5KB/s   00:00
MAINT.FTPcommands                                      100%   88     0.1KB/s   00:00
AUTOLOG1.FTPcommands                                   100%   45     0.0KB/s   00:00
getVMinfo                                              100% 1556     1.5KB/s   00:00
clone.sh                                               100%   16KB  15.8KB/s   00:00
vmQuery                                                100% 9693     9.5KB/s   00:00
clone.hcp.sh                                           100%   16KB  15.6KB/s   00:00
cmsfs-1.1-8c.s390.rpm                                  100%   26KB  25.6KB/s   00:00
```

The files copied are addressed in the sections that follow. Table 7-2 lists a brief description of each file.

Table 7-2 Controller files

File	Description
`/backup/linux/`	An empty directory for Linux backups
`/backup/vm/`	A directory for z/VM configuration file backups
`/usr/local/sbin/clone.sh`	The clone script
`/usr/src/packages/RPMS/s390/cmsfs-1.1-8c.s390.rpm`	The CMS file system RPM
`/etc/Bastille/config`	A Bastille configuration file
`/etc/ntp.conf.template`	An NTP configuration file

File	Description
`/etc/cron.daily/backup_linux`	A script to backup virtual servers
`/etc/cron.daily/backup_vm`	A script to backup important z/VM files
`/etc/cron.daily/*.FTPcommands`	A script to backup important z/VM files
`/etc/ntp.conf.template`	An NTP configuration file

7.6.3 Removing unnecessary RPMs

It is recommended that you remove the following RPMs as you did on the master image:

```
# rpm -e eject fvwm2 hfsutils initviocons ntfsprogs pmtools providers unclutter usbutils
wol xfsprogs zsh
```

You can choose to remove more or fewer RPMs.

7.6.4 Adding additional RPMs

It is recommended that you add the same packages as you did on the master image. Again use the **`yast -i`** command to install these packages:

```
# yast -i findutils-locate bastille xntp rsync sitar wget openmotif nfs-utils
```

You can choose to add more or fewer RPMs.

7.6.5 Installing the cmsfs package

The **`clone.sh`** script requires the **cmsfs** package by Rick Troth in order read CMS files. When you copied files from the NFS server, you should have copied the RPM to the file `/usr/src/packages/RPMS/s390/cmsfs-1.1-8c.s390.rpm`. To install it, run the following:

```
# cd /usr/src/packages/RPMS/s390
# rpm -i cmsfs-1.1-8c.s390.rpm
```

To test that the `cmsfs` package is properly installed, see if you can read `SLES9X`'s parameter file. First you need to bring the 191 disk online with the **`chccwdev -e`** command. Then view the DASD that the system knows with the **`lsdasd`** command, as in Example 7-32.

Example 7-32 Viewing the DASD

```
# chccwdev -e 0.0.0191
Setting device 0.0.0191 online
Done
# lsdasd
0.0.0100(ECKD) at ( 94:  0) is dasda : active at blocksize: 4096, 546840 blocks, 2136 MB
0.0.0101(FBA ) at ( 94:  4) is dasdb : active at blocksize: 512, 524288 blocks, 256 MB
0.0.0102(ECKD) at ( 94:  8) is dasdc : active at blocksize: 4096, 54000 blocks, 210 MB
0.0.0103(ECKD) at ( 94: 12) is dasdd : active at blocksize: 4096, 546840 blocks, 2136 MB
0.0.0104(ECKD) at ( 94: 16) is dasde : active at blocksize: 4096, 540540 blocks, 2111 MB
0.0.0105(ECKD) at ( 94: 20) is dasdf : active at blocksize: 4096, 600840 blocks, 2347 MB
0.0.0106(ECKD) at ( 94: 24) is dasdg : active at blocksize: 4096, 600840 blocks, 2347 MB
0.0.0107(ECKD) at ( 94: 24) is dasdh : active at blocksize: 4096, 600840 blocks, 2347 MB
0.0.0108(ECKD) at ( 94: 24) is dasdi : active at blocksize: 4096, 600840 blocks, 2347 MB
0.0.0191(ECKD) at ( 94: 28) is dasdj : active at blocksize: 4096, 54000 blocks, 210 MB
```

Then use the CMFFS utilities. **`cmsfslst`** lists files on a minidisk and **`cmsfscat`** types the contents of a file, as in Example 7-33 on page 117.

Example 7-33 Listing CMFFS utilities

```
# cmsfslst -d /dev/dasdj
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
         DIRECTOR P0 F         64         13          1  1/25/2006 14:57:53
         ALLOCMAP P0 F       4096          2          2  1/25/2006 15:57:53
CHPW52   XEDIT    B1 V         70        180          3  1/24/2006 14:48:15
CPFORMAT EXEC     B1 V         79        231          3  1/24/2006 14:48:15
LABEL520 EXEC     D1 V         73        112          2  1/24/2006 15:27:13
LABEL520 XEDIT    D1 V         71         19          1  1/24/2006 15:17:05
PROFILE  EXEC     B1 V         71         21          1  1/24/2006 14:48:15
SLES9X   EXEC     B1 V         74          9          1  1/24/2006 14:48:15
SLES9X   INITRD   B1 F         80     162128       3167  1/25/2006 12:11:59
SLES9X   KERNEL   B1 F         80      59299        784  1/25/2006 12:08:45
SLES9X   PARMFILE D1 V         73          8          1  1/25/2006 12:42:50
SWAPGEN  EXEC     B1 V         72        358          5  1/24/2006 14:48:15
LINUX01  PARMFILE D1 V         73          8          1  1/25/2006 15:57:53
# cmsfscat -d /dev/dasdj -a sles9x.parmfile
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=lnx4vm IP_ADDR=129.40.178.127 AUTOINSTALL=yes
IP_HOST=lat127.pbm.ihost.com IP_GATEWAY=129.40.178.254
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=255.255.255.0
IP_BROADCAST=129.40.178.255 READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=129.40.106.1
INST_INFO=nfs INST_IP_ADDR=129.40.178.127
INST_IP_DIR=/nfs/sles9x/sles9xsp3root INST_SCREEN=VNC VNC_PASSWORD=lnx4vm
```

7.6.6 Turning off an unneeded service

There are a number of services which are started in a SLES9 minimum system. In order to keep the controller as lean as possible, it is recommended that **`slpd`**, the Service location protocol daemon, be turned off services with the **`chkconfig`** command:

```
# chkconfig slpd off
```

7.6.7 Turning on the NFS server

The NFS server will be used to export the SLES9 install tree. Both the `nfslock` and the `nfsserver` services are needed. Turn on these two services with the **`chkconfig`** command:

```
# chkconfig nfslock on
# chkconfig nfsserver on
```

When you restart the controller, the NFS lock daemon and server will be started.

7.6.8 Turning on the NTP server

It is desirable to have all the system clocks on your Linux images be the same. This can be accomplished by setting up a time server on the controller, and have the master image and therefore all 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 sync 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.

Look for servers with an access policy of `OpenAccess`. You can see a list of stratum-2 public servers on the Web at:

http://ntp.isc.org/bin/view/Servers/StratumTwoTimeServers

Test to see that the server is indeed accessible by running **`ntpdate`**. In the following example, two servers are used:

```
# ntpdate sundial.columbia.edu
20 Jan 14:21:58 ntpdate[16263]: adjust time server 128.59.59.177 offset -0.004494 sec
# ntpdate clock.nyc.he.net
20 Jan 14:22:14 ntpdate[16264]: adjust time server 209.51.161.238 offset 0.008174 sec
```

The **`xntpd`** configuration file is `/etc/ntp.conf`. In the file `ntp.conf.template`, the two lines specifying the local clock settings in the default configuration file are commented out and the two external time sources (`sundial.columbia.edu` and `clock.nyc.he.net`) are added. It is recommended that you backup the original file and copy this file to the real ntp.conf

```
# cd /etc
# cp ntp.conf ntp.conf.orig
# mv ntp.conf.template ntp.conf
```

Verify that the following changes in Example 7-34 have been made to your new `ntp.conf` file:

Example 7-34 Verifying changes

```
# cat ntp.conf
...
## 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 sundial.columbia.edu
server clock.nyc.he.net
...
```

Start the **`xntpd`** server on the controller with the **`rcxntpd`** command:

```
# rcxntpd start
Try to get initial time via NTP from  sundial.columbia.edu clock.nyc.he.net     done
Starting network time protocol daemon (NTPD)                                    done
# chkconfig xntpd on
```

If you immediately run **`ntptrace`**, you might see that your time server is stratum 16, which means that the clock is not set accurately:

```
# ntptrace
localhost: stratum 16, offset 0.000000, synch distance 0.000000
```

After you start **`ntpd`**, you must wait some time for the clock to be adjusted. Use the **`ntpq`** command with the **`peers`** subcommand to show the two NTP servers that were set in the `/etc/ntp.conf` file:

Example 7-35 Using ntpq> peers to display the servers

```
# ntpq
ntpq> peers
```

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
hickory.cc.colu 128.59.39.48      2 u   48   64    1    5.548   -0.449   0.001
 209.51.161.238  .CDMA.          1 u   47   64    1   16.922   -0.493   0.001
ntpq> quit
```

After a few minutes, the **ntptrace** command should show that your system is now a stratum 2 server:

```
# ntptrace
localhost: stratum 2, offset -0.002555, synch distance 0.006114
avi-lis.gw.lightning.net: stratum 1, offset -0.000191, synch distance 0.000000, refid 'CDMA'
```

7.6.9 Enabling the vmcp module

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. This module was first shipped in SLES9 SP3.

Important: If you are working with a distribution older than SLES9 SP3, you will not have the **vmcp** module/command. If so, you will need to use the cpint package written by Neale Ferguson. It allows you to use the **hcp** command to invoke z/VM CP commands. To install it, issue the following command:

```
# yast -i cpint
# cd /usr/local/sbin
# mv clone.sh clone.vmcp.sh
# mv clone.vmcp.sh clone.sh
```

To enable it, edit the file `/etc/sysconfig/kernel` and add **vmcp** to the variable `MODULES_LOADED_ON_BOOT` (Example 7-36).

Example 7-36 Adding the cmcp module

```
# cd /etc/sysconfig
# vi kernel   // add vmcp to MODULES_LOADED_ON_BOOT
## Path:        System/Kernel
## Description:
## Type:        string
## Command:     /sbin/mkinitrd
#
# This variable contains the list of modules to be added to the initial
# ramdisk by calling the script "mk_initrd"
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="jbd ext3"

## 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"
...
```

Save the file and you should be able to issue CP commands via the **vmcp** after your system is rebooted.

7.6.10 Setting system to halt on SIGNAL SHUTDOWN

Again, SLES9 SP3 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, you want your system to halt (shutdown). Change this setting by changing **`shutdown -r`** to **`shutdown -h`** in the /etc/inittab file, as in Example 7-37.

Example 7-37 Halt on SIGNAL SHUTDOWN

```
# cd /etc
# 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.

7.6.11 Turning off the hz_timer

From the existing SSH session, turn the `hz_timer` off so the controller does not consume unnecessary CPU cycles, as in Example 7-38.

Example 7-38 Turning off the hz_timer

```
# vi /etc/init.d/boot.local    // add one line at the bottom
#! /bin/sh
...#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
sysctl -w kernel.hz_timer=0
```

7.6.12 Configure SSH keys

SSH sessions are typically authenticated with passwords typed in from the keyboard. With SSH key-based authentication sessions can be authenticated with 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.

SSH key-based authentication can be set up from the controller (client) to the virtual servers. If the master 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.

Copy the controller’s public key to the master image’s `authorized_keys` file:

```
# cd /etc/ssh
# cp ssh_host_dsa_key.pub /sles9master/root/.ssh/authorized_keys
```

Make a symbolic link from `/root/.ssh/id_dsa` to the controller’s private key:

```
# cd /root/.ssh
# ln -s /etc/ssh/ssh_host_dsa_key id_dsa
```

7.6.13 Hardening the controller

Run Bastille Linux to harden the controller so that it is more secure. Again you should have copied the file `/etc/Bastille/config`.

Run the **`bastille -b`** command and type `accept` to agree with the user agreement, as in Example 7-39.

Example 7-39 Hardening the controller

```
# bastille -b
Copyright (C) 1999-2002 Jay Beale
Copyright (C) 1999-2001 Peter Watkins
Copyright (C) 2000 Paul L. Allen
Copyright (C) 2001-2003 Hewlett Packard Company
...
You must accept the terms of this disclaimer to use
Bastille.  Type "accept" (without quotes) within 5
minutes to accept the terms of the above disclaimer
> accept
...
```

When the bastille command completes, your system should be hardened according to the settings in the file `/etc/Bastille/config`. Again, you should not be concerned if there are error messages.

7.6.14 Rebooting the system

Reboot the system to test the changes:

```
# reboot
```

After your system comes back in a couple of minutes, start a new SSH session to the controller.

7.6.15 Verifying the changes

You are now finished with 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`** (Example 7-40).

Example 7-40 Verifying the changes

```
# vmcp q n
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
SLES9X  -L0004
VSM     - TCPIP
```

Confirm that both of your swap spaces are operational:

Example 7-41

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdb1                             partition       259956  0       42
/dev/dasdc1                             partition       215896  0       42
```

Verify that the `hz_timer` is off.

```
# cat /proc/sys/kernel/hz_timer
0
```

Verify the NFS server is running:

```
# rcnfsserver status
Checking for kernel based NFS server:
running
```

Congratulations! You have installed and configured Linux twice onto the SLES9X user ID. You are now ready to configure NFS on the controller.

8

Configuring NFS on controller

The `SLES9X` user ID is now customized with both a master image and a controller. The controller should have a 9GB logical volume ext2 file system mounted over `/nfs/`. It can now be configured to replace the NFS (PC) server to make the SLES9 install tree and the files associated with this book available via NFS.

The following steps are involved in configuring NFS on the controller:

- 8.1, “Copying files from NFS server to controller” on page 123
- 8.2, “Configuring the NFS server” on page 124
- 8.3, “Changing the YaST install tree location” on page 125

8.1 Copying files from NFS server to controller

In this section you will copy the SLES9 install tree and the files associated with this book from the NFS server to the controller’s logical volume file system mounted over the directory `/nfs/`.

8.1.1 Copying the SLES9 install tree

This section assumes you have already set up the SLES9 install tree on another server as described in 6.2, “Setting up an SLES9 install tree” on page 76. You can copy the tree recursively then configure the NFS server to export it.

Open or continue an SSH session on the controller. Copy the entire SLES9 install tree recursively using the **`scp -rp`** command. You will need to type the root password of the NFS server:

```
# cd /nfs
# ls
.  ..  lost+found
```

This shows that the file system is empty except for the `lost+found/` directory which is where a file system stores damaged files.

Recursively copy the SLES9 install tree from the NFS server to the `/nfs/` directory:

```
# scp -rp <129.40.46.206>:/nfs/sles9x/sles9xsp3root /nfs
```

```
The authenticity of host '129.40.46.206 (129.40.46.206)' can't be established.
RSA key fingerprint is cc:5e:29:0e:9c:c3:8b:3f:1b:6a:98:46:fd:df:e2:dc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '129.40.46.206' (RSA) to the list of known hosts.
root@129.40.46.206's password:
media                                                   100%   73     0.1KB/s   00:01
directory.yast                                          100%   24     0.0KB/s   00:00
products                                                100%   25     0.0KB/s   00:00
info.txt                                                100%   13KB  13.1KB/s   00:00
...
```

This step should take about 10-30 minutes depending on network and disk speed. Look at the newly copied directory.

```
# ls
.  ..  lost+found  sles9sp3root
# du -sh sles9sp3root/
7.1G    sles9xsp3root/
```

This shows that it occupies about 7 GB on disk.

8.1.2 Copying the files associated with this book

Now recursively copy the files associated with this book that were untarred to `/nfs/` on the NFS server:

```
# scp -rp <129.40.46.206>:/nfs/virt-cookbook /nfs
Password:
root@129.40.46.206's password:
ntp.conf.template                                       100% 2049     2.0KB/s   00:00
rsyncd.conf                                             100%  247     0.2KB/s   00:00
config                                                  100% 2238     2.2KB/s   00:00
set-clock                                               100%   49     0.1KB/s   00:00
...
```

These two directories should use up more than four fifths of the 9GB in the `/nfs/` logical volume as shown by the **`df -h`** command:

```
# df -h | grep nfs
/dev/mapper/system-nfs    9.0G  7.1G  1.5G  83% /nfs
```

8.2 Configuring the NFS server

Set up the NFS server to export the two directories that you just copied. Add two lines to the `/etc/exports` file as follows:

```
# cd /etc
# vi exports     // add a line at the bottom
# 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/virt-cookbook    *(ro,sync)
/nfs/sles9xsp3root    *(ro,sync)
```

The NFS server was set to start when you configured the controller. Double-check that it is running with the **`rcnfsserver`** command:

```
# rcnfsserver status
```

```
Checking for kernel based NFS server:                                                running
```

You can now force the `/etc/exports` file to be reread with the command **`exportfs -a`**:

```
# exportfs -a
```

The NFS server should now be running and customized. Test that you can mount the file systems locally over an empty directory such as `/mnt/` using the DNS name `localhost`, with the following **`mount`** commands:

```
# ls /mnt
# mount localhost:/nfs/virt-cookbook /mnt
# ls -F /mnt
./  ../  README.txt  linux-controller/  linux-master/  nfs-server/  virt-cookbook.pdf
vm/
# umount /mnt
# mount localhost:/nfs/sles9xsp3root /mnt
# ls -F /mnt
./   boot/    control.xml  driverupdate  media.1/  sles9/  yast/
../  content  core9/       linux/        s390x/    sp3-9/
# umount /mnt
```

This shows that you can mount the NFS-exported directories locally.

8.3 Changing the YaST install tree location

When you installed both the master image and the controller, you set the location of the SLES9 install tree to the PC NFS server. Now that you have configured an NFS server on the controller, you can reset this location on both the master image and the controller. After doing this you will no longer need the NFS server as you will be able to use the controller to serve the SLES9 install tree and the files associated with this book.

The steps in this section are as follow:

- "Changing source of installation on master image" on page 125
- "Changing source of installation on controller" on page 127
- "Changing source of installation in parameter file" on page 128

8.3.1 Changing source of installation on master image

There is a bit of a paradox in getting the master image to point to the controller for its source of installation. You have to shutdown the controller to bring up the master image, but if the controller is shut down, **`yast`** will not be able to find the NFS-exported directory.

To work around this issue, you can use the **`chroot`** command which creates a shell with a new root. After **`chroot`**ing you will be working on the running controller, but you will be able to modify the master image because that becomes your new root file system. This technique can be used to perform other modifications to the master image, such as adding RPMs (however, this technique does not work for all commands).

1. Enter the **`chroot`** command to the `/sles9master/` directory and it will be as though you have booted the master image:

```
# chroot /sles9master/
```

2. Now you are in a shell as though you were logged into the master image. Enter the **`yast`** command to change the source of installation.

```
# yast
```

3. Use the Tab key to move to the right side of the YaST Control Center. Use the Down Arrow key twice to select **Change Source of Installation**. Press **Enter**.

```
+------------------------------------------------------------------------------+
¦                             YaST Control Center                              ¦
+------------------------------------------------------------------------------+

+----------------------+ +------------------------------------------------------+
¦Software              ¦ ¦Online Update                                         ¦
¦Hardware              ¦ ¦Install and Remove Software                           ¦
¦System                ¦ ¦Change Source of Installation                         ¦
¦Network Devices       ¦ ¦Installation into Directory                           ¦
¦Network Services      ¦ ¦Patch CD Update                                       ¦
¦Security and Users    ¦ ¦System Update                                         ¦
¦Misc                  ¦ ¦YOU Server Configuration                              ¦
¦                      ¦ ¦                                                      ¦
¦                      ¦ ¦                                                      ¦
¦                      ¦ ¦                                                      ¦
¦                      ¦ ¦                                                      ¦
¦                      ¦ ¦                                                      ¦
+----------------------+ +------------------------------------------------------+

 [Help]                                                                  [Quit]
```

4. You will be presented with the *Software Source Media* panel shown below. You should see three install sources: Service Pack 3, SLES9 and CORE9. These are the sources used to install the master image.
   a. Use the Tab key to move to **Edit** button and press Enter.
   b. Choose the default **Replace...** button.
   c. This will bring up a small menu entitled *NFS Server and Directory*. Change the TCP/IP address (or DNS name) to that of the controller. Change the directory to:

      `nfs/sles9xsp3root/sles9/CD1`

      > **Note:** For a 31-bit system, the three directories will be:
      >
      > ```
      > nfs/sles9sp3root/sles9/CD1
      > nfs/sles9sp3root/core9/CD1
      > nfs/sles9sp3root/sp3-9/CD1
      > ```

   d. Use Tab key to move to **OK** and press Enter.
   e. Repeat this process for the other one or two entries (**Note**: the entries move up as you change them, so you will only have to choose **Edit** one or two more times). Change the remaining two directories to:

      ```
      nfs/sles9xsp3root/core9/CD1
      nfs/sles9xsp3root/sp3-9/CD1
      ```

   f. When you have the desired installation sources, select **Finish**.

.

```
+----------------------+ Software Source Media
¦   Software packages  - +-------------------------------------------------------+
¦can be installed from ¦ ¦Status¦Name                      ¦URL                  ¦
¦the CD, over a network,¦ ¦On    ¦SUSE SLES 9 Service-Pack 3¦nfs://129.40.46.206/n¦
¦or from the hard disk. ¦ ¦On    ¦SUSE SLES Version 9       ¦nfs://129.40.46.206/n¦
¦   To install packages¦ ¦On    ¦SUSE CORE Version 9       ¦nfs://129.40.46.206/n¦
¦from CD, have the     ¦ ¦                                                       ¦
¦&product; CD set or the¦ ¦                                                       ¦
¦DVD available.        ¦ ¦                                                       ¦
¦   The &product; CDs  ¦ ¦                                                       ¦
¦can be copied to the  ¦ ¦                  +------------------------+           ¦
¦hard disk. Then use   ¦ ¦                  ¦NFS Server and Directory¦           ¦
¦that as the           ¦ ¦                  ¦129.40.178.127          ¦           ¦
¦installation source.  ¦ ¦                  ¦Directory on Server     ¦           ¦
¦Insert the path name  ¦ ¦                  ¦/nfs/sles9xsp3root/sp1-9¦           ¦
¦where the first CD is - ¦                  ¦[OK]            [Cancel]¦           ¦
¦located, for example, ¦ ¦                  +------------------------+           ¦
¦/data1/CD1. Only the  ¦ ¦ +----------+                                          ¦
¦base path is required ¦ ¦ ¦Replace...¦                                          ¦
¦if all CDs are copied ¦ ¦ ¦Refresh...¦                                          ¦
¦into one directory.   ¦ ¦ +----------+                                          ¦
¦   Network installation¦ ¦                                                       ¦
¦requires a working    ¦ ++--------------------------------¦---------------------+
¦network connection.   ¦ [      Add      -]          [             Up            ]
¦Configure YaST's      ¦ [     Edit      -]          [           Down            ]
¦"Network/Base" module ¦ [     Delete     ]          [     Enable or Disable     ]
+----------------------+                          [Abort]                 [Finish]
```

5. From the **`yast`** main menu select **Quit**.
6. Use the **`exit`** command to leave the **`chroot`**ed system.

```
# exit
```

Now any Linux virtual server that is cloned should point to the controller for its source of RPMs.

8.3.2 Changing source of installation on controller

Perform the same tasks for the controller, but rather than using NFS as the protocol, just point to the local directories. To do this, you must add 3 new installation sources then delete the existing 3. Invoke **`yast`** and perform the following steps:

1. Choose **Software** => **Change Source of Installation** on the *YaST Control Center*.
2. On the *Software Source Media* panel, choose **Add**.
3. Choose **Local Directory** as shown in the top half of the following example.
4. Set the directory to:

```
/nfs/sles9xsp3root/sles9/CD1
```

5. Repeat steps 2-4 and add an two more entries for:

```
/nfs/sles9xsp3root/core9/CD1
/nfs/sles9xsp3root/sp3-9/CD1
```

6. You should now have six entries: three old entries at the top with `nfs:` prefixes and three new entries below with a `dir:` prefixes. Now delete the three old entries:

   a. Tab to **Delete** and press **Enter**
   b. Verify by accepting **Yes**.
   c. Delete the other two entries. When you are done, the entries should be the same or similar to those shown in the bottom half of the example.
7. Tab to **Finish** and press Enter.

```
+------------------+
¦FTP...            ¦
¦HTTP...           ¦
¦Samba...          ¦
¦NFS...            ¦
¦CD...             ¦
¦DVD...            ¦
¦Local Directory...¦
+------------------+

Software Source Media
+----------------------------------------------------------------+
¦Status¦Name              ¦URL                                    ¦
¦On    ¦SUSE SLES Version 9¦dir:///nfs/sles9xsp3root/sles9/CD1   ¦
¦On    ¦SUSE CORE Version 9¦dir:///nfs/sles9xsp3root/core9/CD1   ¦
¦On    ¦SUSE CORE Version 9¦dir:///nfs/sles9xsp3root/sp3-9/CD1   ¦
```

8.3.3 Changing source of installation in parameter file

There is one more place that the source of the installation has to be changed and that is the parameter file on the `LNXMAINT 192 disk`. Logon to `LNXMAINT` and edit the `SLES9X PARMFILE`. Change the value of the `INST_IP_ADDR` and `INST_IP_DIR` variables. In the following example it is set to `129.40.178.127`:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=lnx4vm IP_ADDR=129.40.178.127 AUTOINSTALL=yes
IP_HOST=lat40.pbm.ihost.com IP_GATEWAY=129.40.178.254
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=255.255.255.0
IP_BROADCAST=129.40.178.255 READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=129.40.106.1
INST_INFO=nfs INST_IP_ADDR=<129.40.178.127>
INST_IP_DIR=/nfs/sles9xsp3root INST_SCREEN=VNC VNC_PASSWORD=lnx4vm
```

Changing this value is somewhat of a paradox. The parameter file sets both the install server IP address and the Linux IP address to the same value. This will never work as you cannot install onto a system that is also the installation server. However, the sections that follow have you copy this parameter file, so this way it will be copied with the correct value.

If for some reason you do have to reinstall onto `SLES9`, you will have to set up another source of installation (perhaps the PC NFS server again), and change this value back to point to it.

8.4 Creating a SLES9 31-bit controller

At this point now you should have completed the following:

- Installed and customized z/VM 5.2

- Set up an NFS server on a PC Linux (or other) box
- Installed Linux twice on the `SLES9X` user ID: once for the master image and once for the controller
- Copied the install trees and files associated with this book from the PC Linux server to the controller.

You should now be ready for cloning in the next chapter.

However, you can choose to repeat the entire process starting with Chapter 7., “Installing and configuring Linux” on page 85 and in this chapter to create a 31-bit controller. Doing so would allow you to clone either 31-bit or 64-bit Linux virtual servers. If you choose to do so, follow the same steps but remove the “X” from the many places where it signifies a 64-bit distribution (e.g. `SLES9X` becomes `SLES9`, `/nfs/sles9xsp3root/` becomes `/nfs/sles9sp3root/`).

Following is an example of a user directory definition for the new ID, named SLES9. This example uses seven more 3390-3s at device addresses `A77A-A780`:

```
*
USER SLES9X LNX4VM 256M 1G BEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3038 <VMA781> MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 <VMA781> MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
 MDISK 103 3390 0001 3338 <VMA782> MR LNX4VM LNX4VM LNX4VM
 MDISK 104 3390 0321 3018 <VMA783> MR LNX4VM LNX4VM LNX4VM
 MDISK 105 3390 0001 3338 <VMA784> MR LNX4VM LNX4VM LNX4VM
 MDISK 106 3390 0001 3338 <VMA785> MR LNX4VM LNX4VM LNX4VM
 MDISK 107 3390 0001 3338 <VMA786> MR LNX4VM LNX4VM LNX4VM
 MDISK 108 3390 0001 3338 <VMA787> MR LNX4VM LNX4VM LNX4VM
*
USER SLES9  LNX4VM 256M 1G BEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3038 <VMA77A> MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 <VMA77A> MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
 MDISK 103 3390 0001 3338 <VMA77B> MR LNX4VM LNX4VM LNX4VM
 MDISK 104 3390 0001 3338 <VMA77C> MR LNX4VM LNX4VM LNX4VM
 MDISK 105 3390 0001 3338 <VMA77D> MR LNX4VM LNX4VM LNX4VM
 MDISK 106 3390 0001 3338 <VMA77E> MR LNX4VM LNX4VM LNX4VM
 MDISK 107 3390 0001 3338 <VMA77F> MR LNX4VM LNX4VM LNX4VM
 MDISK 108 3390 0001 3338 <VMA780> MR LNX4VM LNX4VM LNX4VM
```

Remember to add the new disks to the `$ALLOC$` dummy user ID, so gaps are not reported.

8.5 Retire the PC NFS server

You have now copied all files related to this project from a PC (or other platform) server to zSeries. You should be in a position to retire your PC NFS server.

9

Configure Linux for cloning

The `SLES9` user ID is now customized with both a master image and a controller. The controller should now be running. In this chapter, you will perform the following steps:

- “Defining a new user ID for an virtual server” on page 132
- “Cloning one new virtual server” on page 134
- “Cloning six more virtual servers” on page 137

9.1 Defining a new user ID for an virtual server

In this section you define a new user ID, `LINUX01`, in z/VM and clone the master image to it.

1. Logon to `MAINT` and edit the `USER DIRECT` file to add more Linux ID's.

```
==> x user direct c
```

2. Go to the bottom of the file and add the following five lines in Example 9-1. In this example the user ID will be `LINUX01` with a password of `LNX4VM`. A single 3390-3 DASD is used for a 3038 cylinder (about 2GB) root file system and a 300 cylinder (about 210MB) swap space. In this example it is at device address `A788` which was formatted and given a label of `VMA788` earlier:

Example 9-1 Adding LINUX01 user ID

```
USER LINUX01 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3038 <VMA788> MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 <VMA788> MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
```

3. Add the new volume to the `$ALLOC$` user ID so cylinder 0 will not show up in the disk map as a gap (Example 9-2). Save your changes with the **`FILE`** subcommand:

Example 9-2 Adding the new volume to $ALLOC$

```
====> top
====> /alloc
USER $ALLOC$  NOLOG
MDISK A01 3390 000 001 VVA770 R

...
 MDISK A0B 3390 000 001 <VMA788> R
====> file
```

4. Again check for gaps and overlaps (Example 9-3). You can use the **`ALL`** subcommand with the logical OR operator "|" to check for both strings. You should see only one 501 cylinder gap.

Example 9-3 Checking for gaps and overlaps

```
==> 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 2.0
EOJ DIRECTORY UPDATED AND ON LINE
```

The new Linux user ID has now been defined.

9.1.1 Adding LINUX01 to AUTOLOG1's 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 `AUTOLOG1`'s **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 **XAUTOLOG** the Linux user IDs, as in Example 9-4.

Example 9-4 Linking and accessing the AUTOLOG1 191

```
==> 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 DTCVSW1'             /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'             /* start VSWITCH controller 2 */
'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 180'      /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant sles9x'
'cp set vswitch vsw1 grant linux01'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog sles9x'
'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
```

9.1.2 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 SLES9 manually.

1. Logon to `LNXMAINT`.
2. Copy an existing parameter file and edit the new file to apply to the new Linux.

```
==> copy sles9x parmfile d linux01 parmfile d
==> x linux01 parmfile d
```

3. Edit the new parameter file as you did for `SLES9` (see 7.2, "Preparing SLES9 bootstrap files" on page 88). 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

Example 9-5, the IP address is set to `129.40.178.131` and the DNS name to `lat131.pbm.ihost.com`:

Example 9-5 Editing the new parameter

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=lnx4vm IP_ADDR=<129.40.178.131> AUTOINSTALL=yes
IP_HOST=<lat131.pbm.ihost.com> IP_GATEWAY=<129.40.178.254>
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=<255.255.255.0>
IP_BROADCAST=<129.40.178.255> READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=<129.40.106.1>
INST_INFO=nfs INST_IP_ADDR=<129.40.178.127>
INST_IP_DIR=/nfs/sles9x/sles9xsp3root INST_SCREEN=VNC VNC_PASSWORD=lnx4vm
```

4. Logoff of `LNXMAINT`.
5. Logon to `LINUX01`.
6. Answer **N** to the question "`Do you want to IPL Linux from DASD 103`". Verify that the new Linux user ID has a NIC at addresses 600-602:

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
00: built on IBM Virtualization Technology
...
```

7. Verify that the minidisks at addresses `100` and `102` and the VDISK at address `101` are read/write, as in Example 9-6.

Example 9-6 Verifying that the minidisks are read/write

```
==> q da
00: DASD 0100 3390 VMA788 R/W      3038 CYL ON DASD E34B SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W    524288 BLK ON DASD VDSK SUBCHANNEL = 000E
00: DASD 0102 3390 VMA788 R/W       300 CYL ON DASD E34B SUBCHANNEL = 0001
00: DASD 0190 3390 520RES R/O       107 CYL ON DASD A770 SUBCHANNEL = 0009
00: DASD 0191 3390 VMA77C R/O       300 CYL ON DASD A77C SUBCHANNEL = 000C
...
```

8. Logoff `LINUX01`.

You should now be ready to clone to this new user ID.

9.2 Cloning one new virtual server

Start an SSH session 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/sbin/clone.sh
```

The script takes one parameter which is the Linux user ID that the master image will be cloned to. That user ID must be logged off. It 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`** via the **`vmcp`** module/command to try to copy the 102 (minidisk swap) then the 100 (master image) disks. If **`FLASHCOPY`** fails, the script falls back to copying the disks via the Linux **`dasdfmt`** and **`dd`** commands. The script then boots the new Linux via the **`XAUTOLOG`** command. It also creates an empty backup directory under `/backup/linux/` and adds the server's public key is added to the controller's `known_hosts` file.

It should take less than a minute to clone with **FLASHCOPY** support and three 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, starting with Example 9-7.

Example 9-7 Cloning with FLASHCOPY support

```
# clone.sh linux01
clone.sh linux01
Invoking CP command: QUERY LINUX01
Setting device 0.0.0191 offline
Done
Setting device 0.0.0191 online
Done
LINUX01  PARMFILE D1 V         65          8          1  1/31/2006 10:01:06

WARNING!!: this will copy 100 and 102 disks to LINUX01 100 and 102
New host name will be:  lat44.pbm.ihost.com
New TCP/IP address will be: 129.40.178.131
Other network data is retrieved from LINUX01 PARMFILE on 191 disk
Are you sure you want to overwrite these disks (y/n): y
```

In the section of output in Example 9-7, the script makes sure the user ID to be cloned to exists and is logged off. It then searches for the correct `PARMFILE` and obtains the necessary networking information. It then asks if you are sure you want to overwrite the disks on the target user ID.

Example 9-8 Alternatiing copying and sleeping with FLASHCOPY

```
Copying 0102 swap space to LINUX01 ...
Invoking CP command: QUERY VIRTUAL 1102
Invoking CP command: LINK LINUX01 0102 1102 MR
Invoking CP command: FLASHCOPY 0102 0 END 1102 0 END
Invoking CP command: DETACH 1102
Copying disk via FLASHCOPY succeeded ...
Sleeping 20 seconds for FLASHCOPY to catch up ...
Copying 0100 root file system to LINUX01 ...
Invoking CP command: QUERY VIRTUAL 1100
Invoking CP command: LINK LINUX01 0100 1100 MR
Invoking CP command: FLASHCOPY 0100 0 END 1100 0 END
Invoking CP command: DETACH 1100
Copying disk via FLASHCOPY succeeded ...
```

In Example 9-8, the script copies the 102 minidisk (swap space) to the `LINUX01 102` disk which gets linked as virtual address 1102. It then sleeps for 20 seconds for **FLASHCOPY** to catch up (testing has shown that **FLASHCOPY** cannot perform a second copy to the same volume until the first is done). Then the master image root file system is copied to the `LINUX01 100` minidisk. In both cases, **FLASHCOPY** succeeds. Should it fail, the code will fall back to the **dasdfmt** and **dd** commands to perform the copies.

Example 9-9 Mounting and linking the cloned system

```
Mounting newly cloned image over /mnt/sles9cloned ...
Invoking CP command: LINK LINUX01 100 1100 MR
Setting device 0.0.1100 online
Done
Mounting /dev/dasdk1 over /mnt/sles9cloned ...
Modifying cloned image under /mnt/sles9cloned ...
Regenerating SSH keys in /mnt/sles9cloned/etc/ssh/ ...
Adding 129.40.178.131 to known_hosts file
Setting device 0.0.1100 offline
```

```
Done
Invoking CP command: DETACH 1100
DASD 1100 DETACHED
```

In Example 9-9, the newly cloned file system (`LINUX01 100`) is linked, activated and mounted over a temporary directory `/mnt/sles9cloned/`. Then the networking information is modified in files such as `/etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.0600` and `/etc/HOSTNAME`. Then the SSH keys are regenerated so they are unique for the new virtual server. Then the server's public key is added to the controller's `/root/.ssh/known_hosts` file. Finally the new disk is set offline and detached (Example 9-10).

Example 9-10 Detaching the new disk

```
Invoking CP command: XAUTOLOG LINUX04
Command accepted
Creating a directory under /backup/linux
Successfully cloned /sles9master to LINUX01
You should be able to ping 129.40.178.131 within one minute
```

In the final section, the target user ID is logged on via **`XAUTOLOG`**. Because the `PROFILE EXEC` detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. A directory with the user ID and IP address in the name is created under `/backup/linux/`. The new clone should be on the network in about 30-45 seconds.

Note: If the **`clone.sh`** script fails, you can also add the **`-v`** flag for some more diagnostics. Also, check that:

- The target user ID has been granted access to the VSWITCH.
- The parameter file is copied and set correctly on `LNXMAINT 192`.

A block diagram of this process is displayed n Figure 9-1.

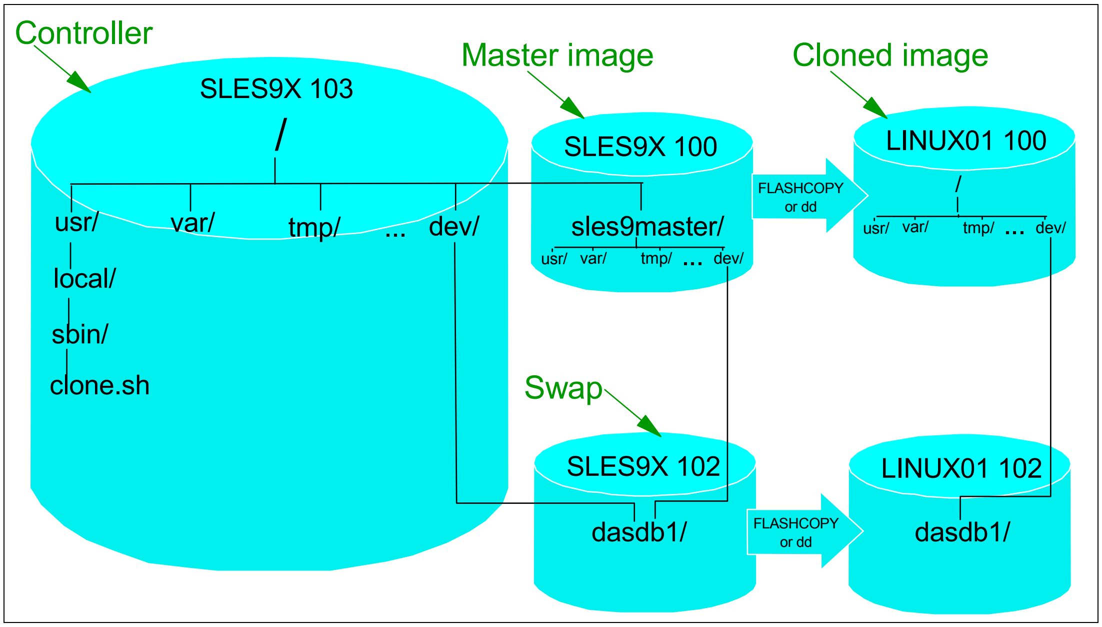


Figure 9-1 Cloning block diagram

The left side of the figure shows the controller which is on the `SLES9X 103` disk. It has the master image mounted over the directory `/sles9master/`, which is the `SLES9 100` disk. Both the controller and the master image use the same minidisk-based swap space, `/dev/dasdb1`, which is the `SLES9 102` disk. Note that the VDISK-based swap space, `SLES9 101`, is created in memory, so it does not need to be copied.

The script **`/usr/local/sbin/clone.sh`** is invoked and it uses either **`CP FLASHCOPY`** or the Linux **`dd`** command to copy the 100 and 102 minidisks to the target z/VM user ID. The script then mounts the newly copied 100 disk and modifies the networking information to use those values found in the parameter file on the `LNXMAINT 192` disk. The script then invokes the **`CP XAUTOLOG`** command to log that user ID on. Because the user ID is logged on disconnected, the common **`PROFILE EXEC`** from the `LNXMAINT 192` disk IPLs from virtual device address 100 and the newly cloned Linux system is brought to life.

9.3 Cloning six more virtual servers

So far you have installed Linux manually twice on `SLES9` to create a master image and a controller. You have created a new user ID `LINUX01` and cloned it. Now it is time to clone six more times to have one system for each of the virtual servers described in the remaining chapters.

The following steps are involved:

- 9.3.1, “Formatting and label six new DASD” on page 137
- 9.3.2, “Defining six more user IDs” on page 138
- 9.3.5, “Testing logging on to a new user ID” on page 142
- 9.3.3, “Creating six new parameter files” on page 140
- 9.3.4, “Granting user IDs access to VSWITCH” on page 141

9.3.1 Formatting and label six new DASD

Decide which DASD will be used for the six new user IDs by referring to 2.5.2, “z/VM DASD worksheet” on page 15. **Logon to MAINT**. In Example 9-11, the devices are `A789-A78E`.

1. Query the devices that you want to assign as PERM space.

Example 9-11 Querying devices for PERM space

```
==> q <a789-a78e>
DASD A789 LAA789  , DASD A78A LAA78A  , DASD A78B LAA78B  , DASD A78C LAA78C
DASD A78D LAA78D  , DASD A78E LAA78E
==> det A789-a78e system
DASD A789 DETACHED SYSTEM
DASD A78A DETACHED SYSTEM
...
```

2. Attach the six DASD to `MAINT`. When attaching volumes to your own user ID, the * parameter can be used.

```
==> att <a789-a78e> *
A789-A78E ATTACHED TO MAINT
```

3. Now format the DASD for PERM or minidisk space with the **`CPFORMAT`** command that is associated with this book.

Example 9-12 formatting the DASD PERM with CPFORMAT

```
==> cpformat <A789-a78e> as perm

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

```
MAINT    A789 MAINT    A789 3390  LAA789 A789         0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A78A MAINT    A78A 3390  LAA78A A78A         0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A78B MAINT    A78B 3390  LAA78B A78B         0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A78C MAINT    A78C 3390  LAA78C A78C         0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A78D MAINT    A78D 3390  LAA78D A78D         0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A78E MAINT    A78E 3390  LAA78E A78E         0       3339

ARE YOU SURE you want to format the DASD as PERM space (y/n)?
y
...
A789-A78E DETACHED
A789-A78E ATTACHED TO MAINT

DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A789 MAINT    A789 3390  VMA789 A789         0       3339
MAINT    A78A MAINT    A78A 3390  VMA78A A78A         0       3339
MAINT    A78B MAINT    A78B 3390  VMA78B A78B         0       3339
MAINT    A78C MAINT    A78C 3390  VMA78C A78C         0       3339
MAINT    A78D MAINT    A78D 3390  VMA78D A78D         0       3339
MAINT    A78E MAINT    A78E 3390  VMA78E A78E         0       3339
```

4. You could now shutdown and re-IPL for these new DASD to be attached to the system with the `User_Volume_Include VM*` statement in the `SYSTEM CONFIG` file. However, there is an easier way. Simply **`DETACH`** the DASD from `MAINT` and **`ATTACH`** them to `SYSTEM`, as in Example 9-13.

Example 9-13 Using DETACH from MAINT and ATTACH to SYSTEM

```
==> det A789-a78e *
A789-A78E DETACHED BY MAINT
==> att A789-a78e system
DASD A789 ATTACHED TO SYSTEM VMA789
DASD A78A ATTACHED TO SYSTEM VMA78A
DASD A78B ATTACHED TO SYSTEM VMA78B
DASD A78C ATTACHED TO SYSTEM VMA78C
DASD A78D ATTACHED TO SYSTEM VMA78D
DASD A78E ATTACHED TO SYSTEM VMA78E
```

The six 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 **`User_Volume_Include VM*`** in the `SYSTEM CONFIG` file.

9.3.2 Defining six more user IDs

Define six more user IDs for Linux virtual servers in the `USER DIRECT` file named `LINUX02 - LINUX07`. You will need to use the DASD volumes you just formatted: one for each virtual server.

1. You can repeat the definition of `LINUX01` six times with the block copy **`""6`** prefix command (Example 9-14).

Example 9-14 Defining six more user IDS

```
==> x user direct
```

```
====> /user linux01
...
""6   *
01846 USER LINUX01 LNX4VM 256M 1G G
01847  INCLUDE LNXDFLT
01848  OPTION APPLMON
01849  MDISK 100 3390 0001 3038 VMA788 MR LNX4VM LNX4VM LNX4VM
01850  MDISK 102 3390 3039 0300 VMA788 MR LNX4VM LNX4VM LNX4VM
""6    MINIOPT NOMDC
```

2. This will create six more copies of the `LINUX01`. Modify them to have a user ID of `LINUX02` - `LINUX07`, and give each new ID the proper 3390-3 identified by label (`VMA789-VMA78E` in tExample 9-15).

Example 9-15   Modifying user IDs and assigning labels

```
USER LINUX02 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3038 VMA789 MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 VMA789 MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
*
USER LINUX03 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3038 VMA78A MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 VMA78A MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
*
...
USER LINUX07 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3038 VMA78E MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 3039 0300 VMA78E MR LNX4VM LNX4VM LNX4VM
 MINIOPT NOMDC
```

3. 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 (Example 9-16).

Example 9-16   Adding dummy definitions for cylinder 0 of each volume

```
====> top
====> /alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 520RES R
...
 MDISK A09 3390 000 001 VMA788 R
 MDISK A0A 3390 000 001 VMA789 R
 MDISK A0B 3390 000 001 VMA78A R
 MDISK A0C 3390 000 001 VMA78B R
 MDISK A0D 3390 000 001 VMA78C R
 MDISK A0E 3390 000 001 VMA78D R
 MDISK A0F 3390 000 001 VMA78E R
...
====> file
```

4. As in Example 9-17, check for overlaps and the single gap. **`Quit`** out of the `USER DISKMAP` file.

Example 9-17 Checking for gaps and overlaps

```
==> 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 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 six new user IDs to which you can clone.

9.3.3 Creating six new parameter files

A new parameter must be created for each of the user IDs with the proper networking information. Link and access the `LNXMAINT 192 disk` read/write and create six new parameter files, as in Example 9-18.

Example 9-18 Creating new parameters for the new user IDs

```
==> link lnxmaint 192 1192 mr
==> acc 1192 f
==> copy linux01 parmfile f linux02 = =
==> copy linux01 parmfile f linux03 = =
==> copy linux01 parmfile f linux04 = =
==> copy linux01 parmfile f linux05 = =
==> copy linux01 parmfile f linux06 = =
==> copy linux01 parmfile f linux07 = =
```

Edit each of the six files by replacing the appropriate network values. For example, in the `LINUX02 PARMFILE` (Example 9-19), only the TCP/IP address and DNS name need to be modified because all other network and other values are the same.

Example 9-19 Editing the six file for network values

```
==> x linux02 parmfile f
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
INST_PASSWORD=lnx4vm IP_ADDR=<129.40.178.132> AUTOINSTALL=yes
IP_HOST=<lat132.pbm.ihost.com> IP_GATEWAY=129.40.178.254
IP_INTERFACE=qeth IP_MTU=1500 IP_NETMASK=255.255.255.0
IP_BROADCAST=129.40.178.255 READ_DEVNO=600 WRITE_DEVNO=601
DATA_DEVNO=602 PORTNAME=dontcare IP_DNS=129.40.106.1
INST_INFO=nfs INST_IP_ADDR=129.40.178.127
INST_IP_DIR=/mnt/sles9root INST_SCREEN=VNC VNC_PASSWORD=lnx4vm
====> file
==> x linux03 parmfile f
...
```

When you are finished, release and detach the `LNXMAINT 192` disk.

```
==> rel f (det
DASD 1192 DETACHED
```

9.3.4 Granting user IDs access to VSWITCH

Modify the `PROFILE EXEC` on `AUTOLOG1 191` to grant access to the VSWITCH for the six 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`**, as in Example 9-20.

Example 9-20 Editing PROFILE EXEC for VSWITCH access

```
==> x profile exec f
...
/* Grant access to VSWITCH for each Linux user */
'CP SET VSWITCH VSW1 GRANT SLES9'
'CP SET VSWITCH VSW1 GRANT LINUX01'
'CP SET VSWITCH VSW1 GRANT LINUX02'
'CP SET VSWITCH VSW1 GRANT LINUX03'
'CP SET VSWITCH VSW1 GRANT LINUX04'
'CP SET VSWITCH VSW1 GRANT LINUX05'
'CP SET VSWITCH VSW1 GRANT LINUX06'
'CP SET VSWITCH VSW1 GRANT LINUX07'

/* XAUTOLOG each Linux user that should be started */
'CP XAUTOLOG SLES9'
'CP XAUTOLOG LINUX01'
'CP XAUTOLOG LINUX02'
'CP XAUTOLOG LINUX03'
'CP XAUTOLOG LINUX04'
'CP XAUTOLOG LINUX05'
'CP XAUTOLOG LINUX06'
'CP XAUTOLOG LINUX07'
====> file
```

It is easiest to grant access to the new user IDs for the current z/VM session with the **`SET VSWITCH`** command (Example 9-21).

Example 9-21 Using SET VSWITCH

```
==> set vswitch vsw1 grant linux02
Command complete
==> set vswitch vsw1 grant linux03
Command complete
==> set vswitch vsw1 grant linux04
Command complete
==> set vswitch vsw1 grant linux05
Command complete
==> set vswitch vsw1 grant linux06
Command complete
==> set vswitch vsw1 grant linux07
Command complete
```

Verify that the user IDs have access with the **`QUERY VSWITCH ACCESSLIST`** command (Example 9-22).

Example 9-22 Verifying access

```
==> query vswitch vsw1 acc
```

```
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 4 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
    Authorized userids:
      SLES9    LINUX01  LINUX02  LINUX03  LINUX04  LINUX05
      LINUX06  LINUX07
      SYSTEM
```

9.3.5 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:

```
LOGON LINUX02
00: NIC 0600 is created; devices 0600-0602 defined
...
```

If you forgot to grant access to the VSWITCH you will see an error message. Verify that you have OSA devices at addresses `600-602`, and read/write DASD devices at addresses `100-102` (Example 9-23).

Example 9-23 Verfying devices on LINUX02 with LINUX07 user ID

```
==> q osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
00:      0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
...
==> q da
00: DASD 0100 3390 VMA789 R/W       3038 CYL ON DASD  E34F SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 0011
00: DASD 0102 3390 VMA789 R/W        300 CYL ON DASD  E34F SUBCHANNEL = 0001
00: DASD 0190 3390 520RES R/O        107 CYL ON DASD  A770 SUBCHANNEL = 0009
...
```

Log off LINUX02.

Congratulations, you have cloned one Linux virtual server and defined six more user IDs that should now be ready for receiving cloned images. You will clone to these user IDs in the sections that follow.

9.4 Reviewing system status

You can step back now and view your system from a DASD point of view as shown in Figure 9-2 on page 143. If you have followed all sections in this book you should have used 24 3390-3 volumes: 10 for your z/VM system, 7 for the Linux controller and master image and one for each of the seven virtual servers.

You can also view the system from an administrator's and end user's point of view, as shown by the horizontal lines and the italicized text on the right side of the figure. The z/VM and Linux system administration roles may be performed by the same person, but these roles can also be done by different administrators. The Linux end users might not care that their servers are virtual machines and can be oblivious to the fact that they might have been cloned in a matter of minutes.

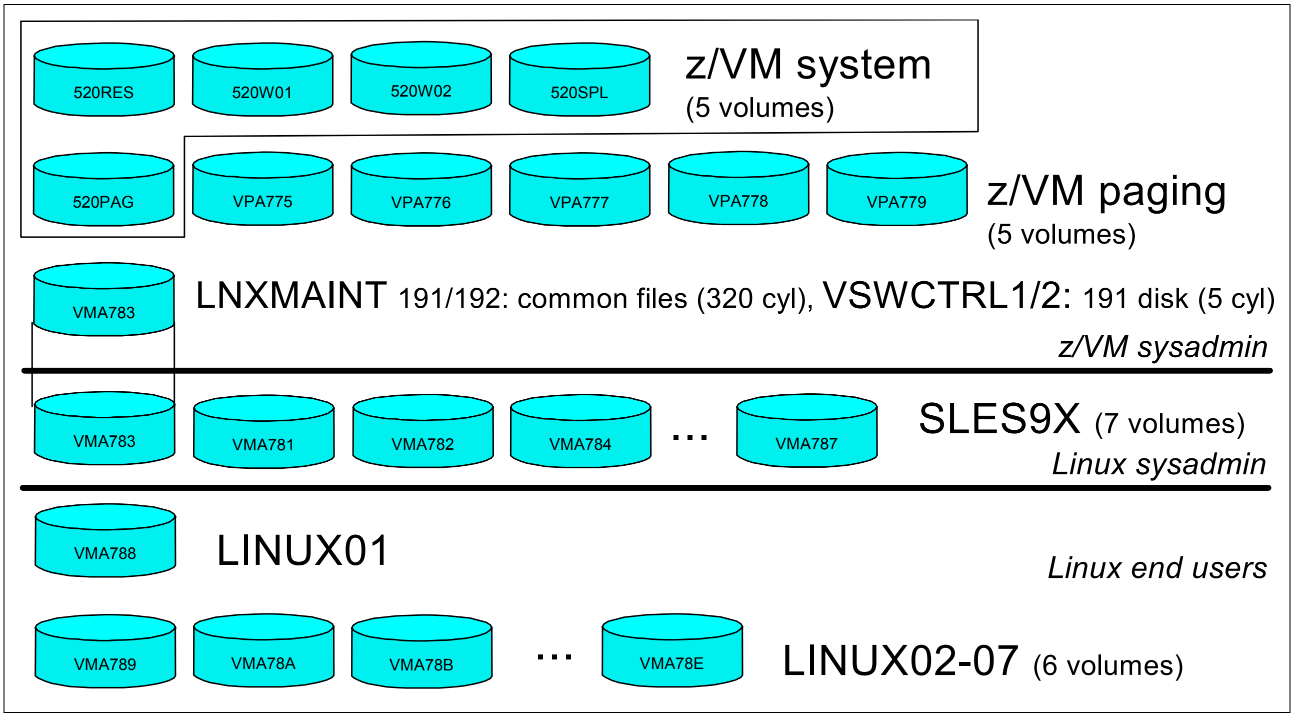


Figure 9-2 Linux virtual server system - DASD view and role view

10

Four virtual servers

This chapter describes how to clone and customize the following Linux virtual servers:

- 10.1, “Creating a virtual Web server” on page 146
- 10.2, “Creating a virtual LDAP server” on page 150
- 10.3, “Creating a virtual file and print server” on page 157
- 10.4, “Creating a virtual application development server” on page 161

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, "Configure Linux for cloning" on page 131.

10.1.1 Installing Apache RPMs

SSH into the IP address of the new `LINUX01` server. Install the following Apache RPMs via the **`yast -i`** command:

```
# yast -i apache2-prefork apache2 apache2-doc apache2-example-pages
```

You will see **`yast`** menus go by as the Apache RPMs are installed. When it is complete you can confirm the RPMs have been added via the **`rpm -qa`** command:

```
# rpm -qa | grep apache
apache2-2.0.49-27.8
apache2-example-pages-2.0.49-27.8
apache2-prefork-2.0.49-27.8
apache2-doc-2.0.49-27.8
```

10.1.2 Testing Apache

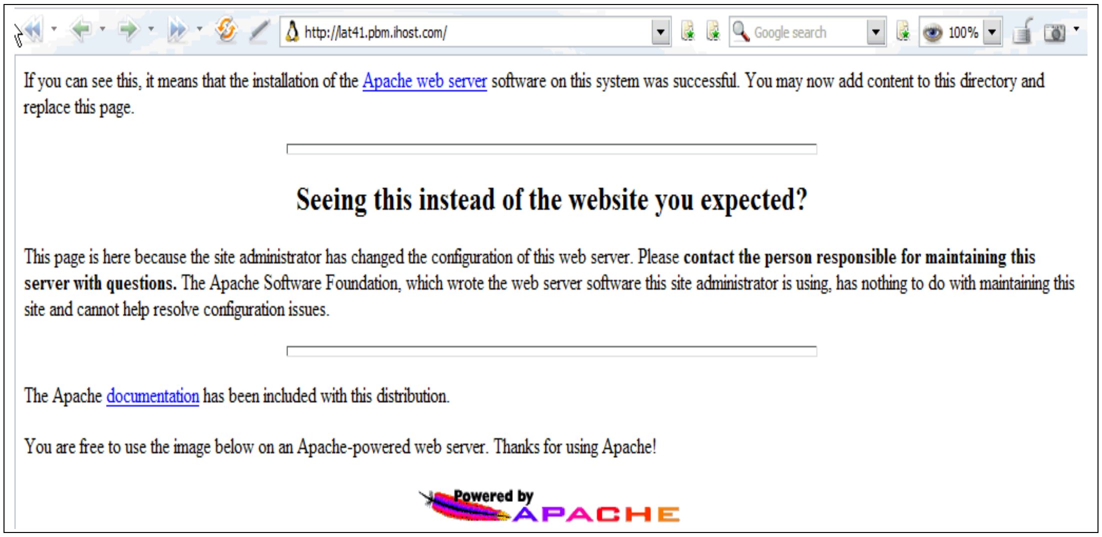
Start the Apache Web server to verify that it is installed successfully. You must start Apache as `root`, but it will then launch child processes as a less privileged user to listen for and handle requests.

```
# rcapache2 start
Starting httpd2 (prefork)                                             done
```

To verify that Apache is installed correctly, after it has been started, point a Web browser to the server and see the Apache test page. In your Web browser, enter 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 lat131.pbm.ihost.com:

```
http://lat131.pbm.ihost.com
```

You should see a test page similar to Figure 10-1.



If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.

Seeing this instead of the website you expected?

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions.** The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache documentation has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

Figure 10-1 Apache2 test page

If you get an error in starting Apache, look in the log file `/var/log/apache2/error-log` for clues. If Apache started successfully but you can't reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

10.1.3 Configuring SSL for Apache

Secure Sockets Layer (SSL) is used to encrypt data between the client (browser) and the server. In order for the client to know you are a legitimate Web server, you will first need to create a server certificate. Then, with your certificate in hand, you can setup Apache to provide secure communications with SSL.

Creating a server certificate

It is recommended that you first create a self-signed certificate to test that your SSL configuration is correct. Then, for production purposes, you might want to purchase a certificate signed by a trusted Certificate Authority (CA).

Use OpenSSL to create the certificate. This process includes these steps.

1. Create a public/private key pair.
2. Create a certificate request.
3. Create a server certificate.
4. Use the **`openssl genrsa`** command to generate a RSA key pair (Example 10-1). The **`-rand`** switch is used to provide OpenSSL with randomized data to ensure that generated keys are unique and unpredictable. Substitute <**`file1:file2:file3>`** with paths to large, random files on your system.

Example 10-1 Generating an RSA key pair

```
# cd ~
# openssl genrsa -rand <file1:file2:file3> -out <lat131.key> 1024
43208 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.......................++++++
.............++++++
e is 65537 (0x10001)
```

5. Create a certificate request (Example 10-2. This is needed to create a self-signed certificate or to obtain a CA-signed certificate. The creation process will ask you questions about your business. Answer them as appropriate:

Example 10-2 Creating a certificate request

```
# openssl req -new -key <lat131.key> -out <lat131.csr>
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:<US>
State or Province Name (full name) [Some-State]:<NY>
Locality Name (eg, city) []:<Poughkeepsie>
Organization Name (eg, company) [Internet Widgits Pty Ltd]:<IHOST>
Organizational Unit Name (eg, section) []:<PBM>
Common Name (eg, YOUR name) []:<Admin>
Email Address []:<admin@pbm.ihost.com>
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:<a3tfgm>
An optional company name []:<IHOST>
```

6. To obtain a server certificate, you can either create a self-signed certificate, or obtain a trusted CA-signed certificate. Trusted Certificate Authorities include GeoTrust, VeriSign and Thawte. Submit your certificate signing request (the `.csr` file you just created) to one of them for processing (See 10.1.5, “Apache resources” on page 150 for their URLs). To create your own self-signed certificate, run the following command in Example 10-3.

Example 10-3 Creating a self-signed certificate

```
# openssl x509 -req -days 30 -in <lat131.csr> -signkey <lat131.key> -out <lat131.crt>
Signature ok
subject=/C=US/ST=NY/L=Poughkeepsie/O=IHOST/OU=PBM/CN=Admin/emailAddress=admin@ihost.pbm.
com
Getting Private key
```

7. Move the SSL files into the appropriate directories under Apache:

```
# mv <lat131.key> /etc/apache2/ssl.key
# mv <lat131.crt> /etc/apache2/ssl.crt
```

Setting up virtual hosts

Because SSL-protected Web pages run on a different port than the non-protected Web pages, you should consider them separate Web servers. A common way of serving an SSL-enabled Web site is to create a *virtual host* on the Web server. Apache’s Virtual Host capability allows you to have multiple Web servers on one machine. The main configuration file `/etc/apache2/httpd.conf` reads virtual hosts information from included configuration files in the directory `/etc/apache2/vhosts.d/`.

1. To create a virtual host configuration file, copy the template file `vhost-ssl.template` to your own configuration file and customize it with the paths to SSL logs and your SSL certificate and key. The lines that need to be modified are in bold in Example 10-4.

Example 10-4 Creating a virtual host configuration file

```
# cd /etc/apache2/vhosts.d
# cp vhost-ssl.template ssl.conf
# vi ssl.conf
...
<VirtualHost _default_:443>

        #  General setup for the virtual host
        DocumentRoot "/srv/www/htdocs"
        #ServerName www.example.com:443
        #ServerAdmin webmaster@example.com
        ErrorLog /var/log/apache2/ssl_error_log
        TransferLog /var/log/apache2/ssl_access_log
...
        #   Server Certificate:
        #   Point SSLCertificateFile at a PEM encoded certificate.  If
        #   the certificate is encrypted, then you will be prompted for a
        #   pass phrase.  Note that a kill -HUP will prompt again.  Keep
        #   in mind that if you have both an RSA and a DSA certificate you
        #   can configure both in parallel (to also allow the use of DSA
        #   ciphers, etc.)
        SSLCertificateFile /etc/apache2/ssl.crt/<lat131.crt>
        #SSLCertificateFile /etc/apache2/ssl.crt/server-dsa.crt

        #   Server Private Key:
        #   If the key is not combined with the certificate, use this
        #   directive to point at the key file.  Keep in mind that if
        #   you've both a RSA and a DSA private key you can configure
        #   both in parallel (to also allow the use of DSA ciphers, etc.)
        SSLCertificateKeyFile /etc/apache2/ssl.key/<lat131.key>
        #SSLCertificateKeyFile /etc/apache2/ssl.key/server-dsa.key
...
```

2. Edit the configuration file for the Apache startup script, `/etc/sysconfig/apache2`, to add a start time flag to let Apache know to enable SSL:

```
# vi /etc/sysconfig/apache2
...
APACHE_SERVER_FLAGS="SSL"
```

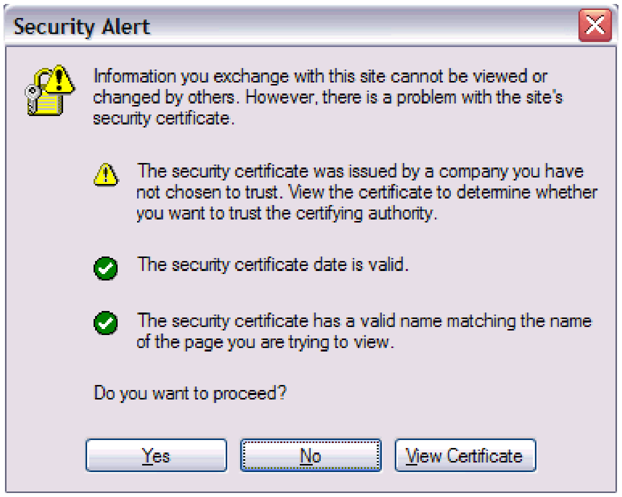
3. If Apache is already running, you need to restart it to take the changes:

```
# rcapache2 restart
Syntax OK
Shutting down httpd2 (waiting for all children to terminate)         done
Starting httpd2 (prefork)                                            done
```

4. Test the SSL-enables Web server by pointing a browser to:

   `https://<lat131.pbm.ihost.com>`

   If you are using a self-signed certificate, then you will see a warning before your browser downloads the page (Figure 10-2). It is simply telling you that it is not signed by a trusted CA, click **Yes** to proceed:



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate date is valid.

The security certificate has a valid name matching the name of the page you are trying to view.

Do you want to proceed?

Figure 10-2 Security warning

You should again see a page similar to that shown in Figure 10-1 on page 146, the only difference being that this one has an `https` prefix, not `http`.

You can customize your Web site with SSL in many ways, such as choosing which pages are SSL enabled and which SSL Ciphers to use, etc. Refer to the Apache documentation in 10.1.5, “Apache resources” on page 150 for more details.

10.1.4 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, i.e. `/etc/apache2/my-server.conf`.

Make the changes in `/etc/apache2/my-server.conf`, and then edit `/etc/apache2/httpd.conf` to use `my-server.conf`.

10.1.5 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 with the OpenLDAP package, which comes standard with most Linux distributions. Among other directory functions, OpenLDAP allows for centralized login authentication and user and group ID resolution.

In this section you install Linux manually and set up login authentication to the new virtual LDAP server. Then you return to the virtual Web server you just created to point it to the new LDAP server.

Then you might want to configure the master 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:

- 10.2.1, “Manually installing a Linux virtual server” on page 150
- 10.2.2, “Configuring the virtual LDAP server” on page 153
- 10.2.3, “Investigating the new virtual server” on page 153
- 10.2.4, “Adding a new user” on page 154
- 10.2.5, “Setting another virtual server to use LDAP server” on page 154
- 10.2.6, “Pointing the master image to the virtual LDAP server” on page 157

10.2.1 Manually installing a Linux virtual server

For OpenLDAP to work properly, you must have a working DNS server and the DNS name and IP address values set in the `LINUX02 PARMFILE` must be associated.

It is recommended that you manually install Linux to create a virtual LDAP server rather than cloning a virtual server. The reason for this is because of the LDAP and certificate configuration module that is invoked in the second half of a manual install. When installing both the master image and controller, it was recommended that you skip the Service Configuration window. When installing the virtual LDAP server, this YaST configuration module will be used. It makes setting up OpenLDAP much easier.

1. Open a 3270 session to `LINUX02` to install Linux manually. If Linux is running you can kill the current system with the command **`#CP IPL CMS`**. You should be prompted to IPL Linux - answer **n** to the question:

```
LOGON LINUX02
...
Do you want to IPL Linux from DASD 100? y/n
n
```

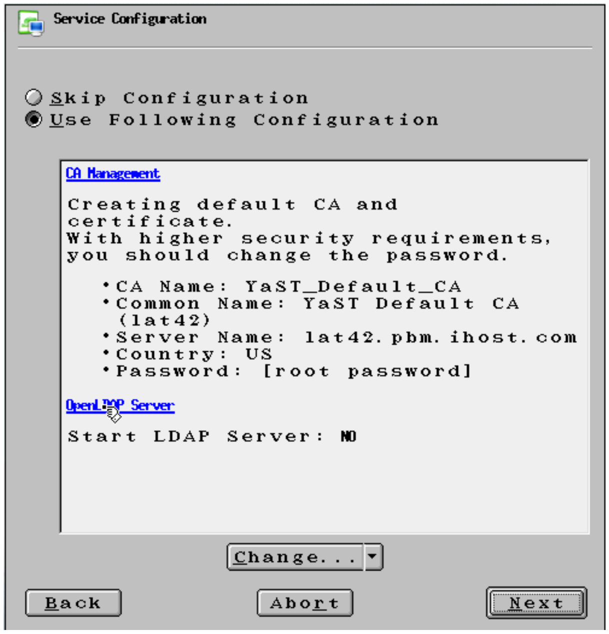
2. Invoke the **`SLES9X EXEC`** to begin a manual install of a 64-bit system

```
==> sles9x
...
```

3. Complete the first half of the install as described in 7.3.1, “Begin the SLES9 installation” on page 92. If you have cloned to `LINUX02` you can safely skip the formatting of disks 100 and 102 after you activate them. The first half of the install will be the same.

From a 3270 session you will **`IPL 100`** to complete the second half of the installation which is described in 7.3.5, “Completing YaST2 installation” on page 97, however there will be some differences:

1. Set the root password.
2. Accept the default values in the Network Configuration panel.
3. On the Test Internet Connection panel, select **No, Skip This Test**.
4. Do not skip the Service Configuration panel. Accept the default of the **Use Default Configuration** radio button and click **OpenLDAP server,** as shown in Figure 10-3.



○ Skip Configuration
● Use Following Configuration

CA Management

Creating default CA and certificate.
With higher security requirements, you should change the password.

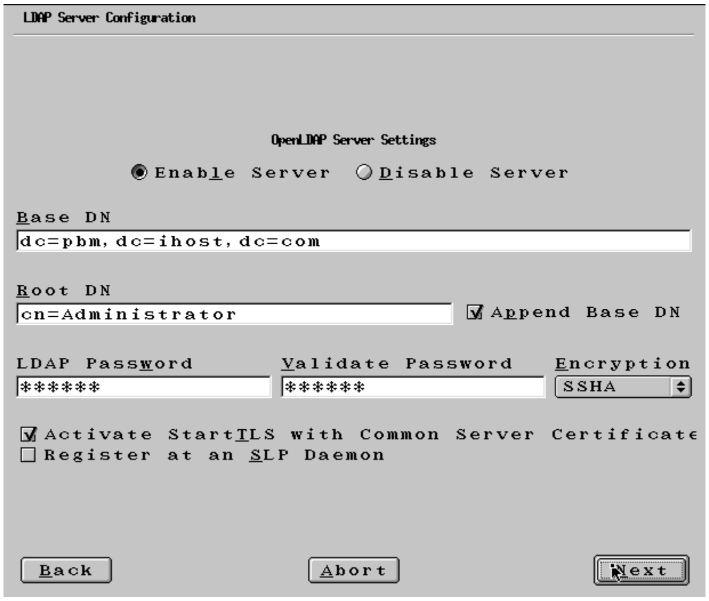
- CA Name: YaST_Default_CA
- Common Name: YaST Default CA (lat42)
- Server Name: lat42.pbm.ihost.com
- Country: US
- Password: [root password]

OpenLDAP Server

Start LDAP Server: NO

Figure 10-3 YaST module to configure OpenLDAP and certificates

5. You will see a warning message starting with Changing anything in this dialog ... Click **OK**.
6. This should bring you to the Configure LDAP Server panel as shown in Figure 10-4 on page 152. Click the **Enable Server** radio button. *Deselect* the **Register at an SLP Daemon** check box at the bottom and click **Next**.


LDAP Server Configuration

OpenLDAP Server Settings

● Enable Server ○ Disable Server

Base DN: dc=pbm, dc=ihost, dc=com

Root DN: cn=Administrator

☑ Append Base DN

LDAP Password: ******

Validate Password: ******

Encryption: SSHA

☑ Activate StartTLS with Common Server Certificate

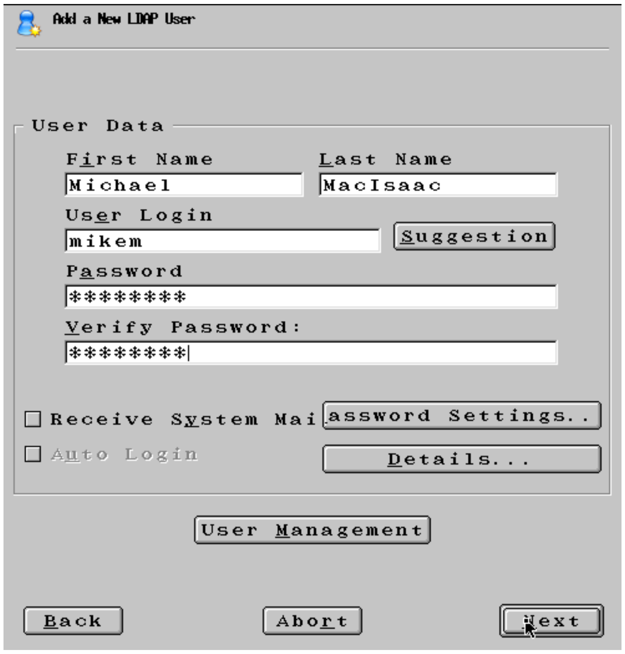
☐ Register at an SLP Daemon

Figure 10-4 LDAP Server Configuration panel

7. This returns you to the Service Configuration panel. Click **Next**.
8. You should see a certificate being created.
9. This take you to a message box stating To configure the LDAP server the OpenLDAP package must be installed. Click **Continue** and the appropriate RPMs will be installed.

Important: If you see the error `Could not create database`, this is not good. This appears to be a result of using the Back button during the process described. If you get this error mssage, it is probably easiest to begin the install again, because LDAP server is in an unknown state.

10. In the User Authentication Method panel accept the default of **LDAP** and click **Next**.
11. You again see a message that the pam_ldap and nss_ldap packages must be installed. Click **Continue** and the appropriate RPMs are installed.
12. In the LDAP Client Configuration panel, accept the defaults and click **Next**.
13. In the Add a new LDAP user panel, add one new LDAP user. An example is shown in Figure 10-5 on page 153 of adding the user `mikem`. Click **Next** when you are ready.



First Name: Michael
Last Name: MacIsaac
User Login: mikem
Password: ********
Verify Password: ********

Figure 10-5 Add New LDAP User panel

14. Accept the defaults on the last two panels of the installation panels and click **Finish**.
15. Go back to your 3270 session and clear the screen. Your VNC session ends and your new system completes booting.

You should now have a vanilla 64-bit SLES9 system with an LDAP server installed and initially configured.

10.2.2 Configuring the virtual LDAP server

Because you did not clone this server from the master image, you did not pick up the configuration that you did to it. Go back to section 7.4, “Configuring the master image” on page 98 and repeat the steps that you performed on the master image.

10.2.3 Investigating the new virtual server

Your new system should come back in a minute or two.

1. Start an SSH session as root to your new virtual server.
2. Verify that LDAP is running:

```
# rcldap status
Checking for service ldap:                                          running
```

3. The user that was added (`mikem` in this example) should exist. Look for it the entry in the `/etc/passwd` file:

```
# id <mikem>
uid=1000(mikem) gid=100(users) groups=100(users)
# grep <mikem> /etc/passwd
```

4. The **`grep`** command gives no output. Why is this? Because the user was not added to the local file system. Rather, it was added to the OpenLDAP database. Confirm this with the **`ldapsearch -x`** command searching for the entry with **`uid=mikem`** in Example 10-5 on page 154:

Example 10-5 Searching for uid=mikem

```
# ldapsearch -x uid=<mikem>
...
# mikem, people, pbm.ihost.com
dn: uid=mikem,ou=people,dc=pbm,dc=ihost,dc=com
...
```

This shows that the user named `mikem` has been added to the LDAP server.

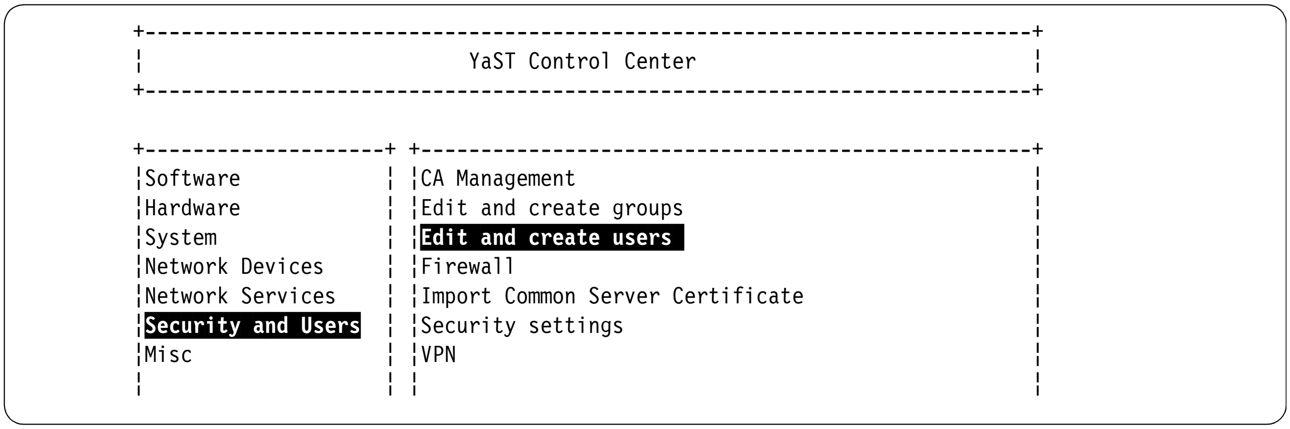
10.2.4 Adding a new user

The first LDAP user was added during Linux installation. There are a number of different ways to adding LDAP users. It can be done with the **`ldapadd`** command and a manually edited LDIF file, or it can be done with a graphical LDAP browser such as **`gq`**. But perhaps **`yast`** in curses mode is a good compromise between these two.

1. Invoke **`yast`**:

```
# yast
```

Figure 10-6 Invoking YaST



```
+-------------------------------------------------------------------------+
¦                           YaST Control Center                           ¦
+-------------------------------------------------------------------------+

+---------------------+ +---------------------------------------------------+
¦Software             ¦ ¦CA Management                                      ¦
¦Hardware             ¦ ¦Edit and create groups                             ¦
¦System               ¦ ¦Edit and create users                              ¦
¦Network Devices      ¦ ¦Firewall                                           ¦
¦Network Services     ¦ ¦Import Common Server Certificate                   ¦
¦Security and Users   ¦ ¦Security settings                                  ¦
¦Misc                 ¦ ¦VPN                                                ¦
¦                     ¦ ¦                                                   ¦
```

2. On the main panel, choose **Security and Users** then **Edit and create users** on the right side.
3. You are prompted for the LDAP (`root`) password. Enter the password, Tab to **OK** and press Enter.
4. You can see the Edit and Create Users panel. Tab to **Add** and press Enter.
5. You can see the User Data panel. Enter the information for the new user, tab to **Create** and press Enter.
6. In the User and Group Administration panel, tab to **Finish** and press Enter. The user will be added to the LDAP server
7. In the YaST Control Center, tab to **Quit** and press Enter.

Now you should be able to SSH into the LDAP virtual server with the new user's credentials.

10.2.5 Setting another virtual server to use LDAP server

Now that you have a virtual LDAP server, you can point the other virtual servers to it so 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. You must install some RPMs and do some configuration. In this section you perform the following steps:

- “Testing that the LDAP client is not working” on page 155
- “Adding two LDAP RPMs” on page 155
- “Using YaST to modify the LDAP authentication client” on page 155
- “Modifying the OpenLDAP client configuration file” on page 156
- “Testing the LDAP client” on page 157

Testing that the LDAP client is not working

Before you start, try a couple of commands to show 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. In this example it is mikem.

```
# ldapsearch -x uid=<mikem>
ldap_bind: Can't contact LDAP server (-1)
# id <mikem>
```

The **`ldapsearch`** command cannot resolve the LDAP user because it cannot contact the LDAP server. Similarly, the **`id`** command gives no output for the same reason.

Adding two LDAP RPMs

Use the **`yast -i`** command to add the RPMs `pam_ldap` and `nss_ldap`:

```
# yast -i pam_ldap nss_ldap
```

You should see the packages being added in YaST curses screens. When the process is complete verify that the two packages were added with the following **`rpm`** command:

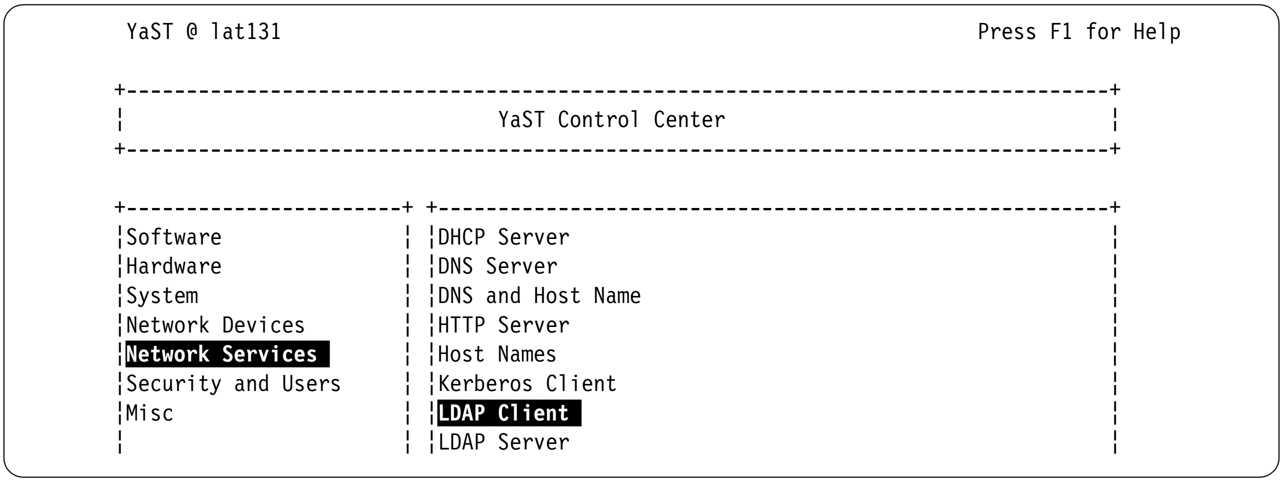
```
# rpm -qa | grep _ldap
pam_ldap-169-28.1
nss_ldap-215-59.5
```

Using YaST to modify the LDAP authentication client

The **`yast`** system administration interface can be used to configure the LDAP authentication client. As in Figure 10-7, select **Network Services** on the left side of the main screen, then LDAP Client:

```
# yast
```

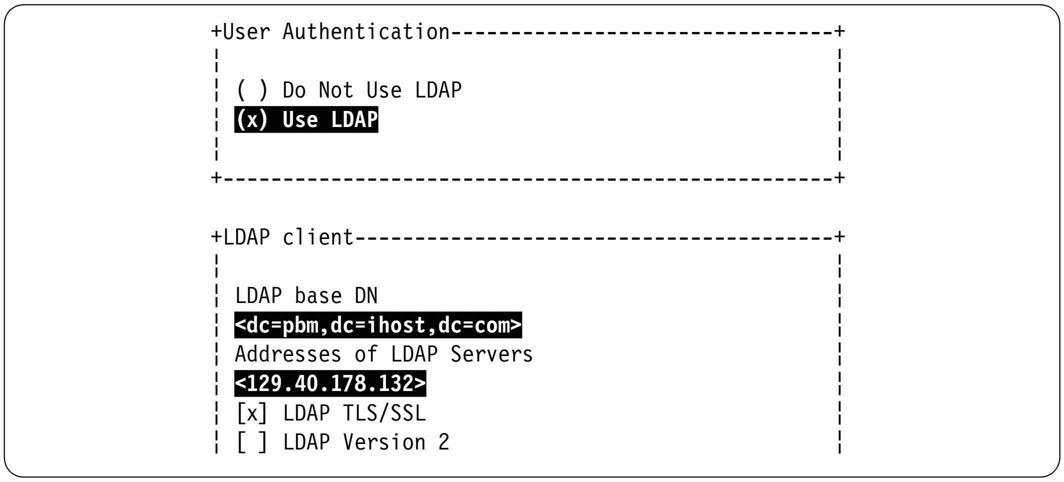
Figure 10-7 Using YaST to configure LDAP authentication client



```
 YaST @ lat131                                                           Press F1 for Help

+-----------------------------------------------------------------------------------+
¦                               YaST Control Center                                 ¦
+-----------------------------------------------------------------------------------+

+-----------------------+ +---------------------------------------------------------+
¦Software               ¦ ¦DHCP Server                                              ¦
¦Hardware               ¦ ¦DNS Server                                               ¦
¦System                 ¦ ¦DNS and Host Name                                        ¦
¦Network Devices        ¦ ¦HTTP Server                                              ¦
¦Network Services       ¦ ¦Host Names                                               ¦
¦Security and Users     ¦ ¦Kerberos Client                                          ¦
¦Misc                   ¦ ¦LDAP Client                                              ¦
¦                       ¦ ¦LDAP Server                                              ¦
```

On the panel that follows in Figure 10-8, use the Tab key to move to **Use LDAP** and press the Space bar to select that choice:

Figure 10-8 User Authentication panel



```
+User Authentication--------------------------------+
¦                                                   ¦
¦ ( ) Do Not Use LDAP                               ¦
¦ (x) Use LDAP                                      ¦
¦                                                   ¦
+---------------------------------------------------+

+LDAP client----------------------------------------+
¦                                                   ¦
¦ LDAP base DN                                      ¦
¦ <dc=pbm,dc=ihost,dc=com>                          ¦
¦ Addresses of LDAP Servers                         ¦
¦ <129.40.178.132>                                  ¦
¦ [x] LDAP TLS/SSL                                  ¦
¦ [ ] LDAP Version 2                                ¦
```

That is all. Use the Tab key to move to **Next** and press Enter, then do the same for the **Quit** button on the main window. The changes will be made.

Modifying the OpenLDAP client configuration file

The previous step modifies the `/etc/ldap.conf` and `/etc/nsswitch.conf` files which are sufficient for LDAP client authentication, however, it does not modify the `/etc/openldap/ldap.conf` file which is used for the **`ldap*`** commands.

You do not need to make a backup as there is already a copy in the file `ldap.conf.default`. You can verify this with the **`diff`** command. If the files are identical there will be no output.

```
# cd /etc/openldap
# ls ldap.*
ldap.conf  ldap.conf.default
# diff ldap.conf ldap.conf.default
```

You might want to rename the file to follow the naming convention:

```
# mv ldap.conf.default ldap.conf.orig
```

Modify the OpenLDAP client configuration file. This is the file that the **`ldap*`** commands read. Set the BASE variable to the suffix of the LDAP tree (`dc=pbm, dc=ihost, dc=com` in this example), and set the URI variable to the LDAP server TCP/IP address (`129.40.178.132` in this Example 10-6):

Example 10-6 Modifying the OpenLDAP client configuration file

```
# vi ldap.conf     // modify the BASE (LDAP tree suffix) and the URI (LDAP server)
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    <dc=pbm, dc=ihost, dc=com>
URI     <ldap://129.40.178.132>
```

Testing the LDAP client

Save the file. Now try the **`id`** and **`ldapsearch`** commands again (Example 10-7). This time they should both succeed:

Example 10-7 Testing the LDAP client

```
# id mikem
uid=1000(mikem) gid=100(users)
groups=100(users),14(uucp),16(dialout),17(audio),33(video)
# ldapsearch -x uid=mikem
# extended LDIF
#
# LDAPv3
# base <> with scope sub
# filter: uid=mikem
# requesting: ALL
#

# mikem, people, pbm.ihost.com
dn: uid=mikem,ou=people,dc=pbm,dc=ihost,dc=com
...
```

You should also be able to start an SSH session to the virtual Web server using the LDAP user.

10.2.6 Pointing the master image to the virtual LDAP server

You can point the master Linux image to this virtual LDAP server. To do so, you could shut down the controller, bring up master image (via **`IPL 100`**), point it to the new LDAP server the same way you just did. However, an easier way is to use the **`chroot`** command. Start an SSH session

```
# chroot /sles9master/
```

Now perform the same steps as you did on the virtual Web server (section 10.2.5, “Setting another virtual server to use LDAP server” on page 154). When you are done, exit form the **`chroot`**ed session:

```
# exit
```

Now clone a virtual server and test logging in as an LDAP user. If this succeeds, all virtual servers cloned will be able to authenticate login sessions to the LDAP user and group directory.

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 CUPS - the Common UNIX Printing System. 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 follow:

- 10.3.1, “Cloning a Linux virtual server” on page 158
- 10.3.2, “Installing necessary RPMs” on page 158
- 10.3.3, “Running Bastille” on page 158

- 10.3.4, “Configuring Samba configuration file” on page 159
- 10.3.5, “Adding a Samba user” on page 159
- 10.3.6, “Starting Samba at boot time” on page 160
- 10.3.7, “Testing your changes” on page 160

10.3.1 Cloning a Linux virtual server

From the controller, clone a basic virtual server. In this example the user ID `LINUX03` is used.

```
# clone.sh linux03
...
```

SSH in to the new virtual server.

10.3.2 Installing necessary RPMs

Add the following RPMs via the **`yast -i`** command:

```
# yast -i samba yast2-samba-server samba-doc samba-pdb samba-vscan samba-winbind cups \
cups-drivers ghostscript-serv
```

You can see YaST curses screens flash by as the RPMs are added to the system.

Confirm that the RPMs were added, as in Example 10-8.

Example 10-8 Confirming added RPMs

```
# rpm -qa | egrep “samba|cups”
cups-libs-1.1.20-108.15
samba-doc-3.0.9-2.1.5
samba-pdb-3.0.9-2.1.5
samba-vscan-0.3.5-11.7.5
yast2-samba-client-2.9.17-1.3
samba-client-3.0.9-2.1.5
cups-client-1.1.20-108.15
samba-3.0.9-2.1.5
samba-winbind-3.0.9-2.1.5
yast2-samba-server-2.9.28-1.7
cups-drivers-1.1.20-66.6
cups-1.1.20-108.15
```

When completed you should still have about 1.4GB free (your values might differ):

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.1G  656M  1.4G  33% /
tmpfs                 124M     0  124M   0% /dev/shm
```

10.3.3 Running Bastille

When Bastille was run after the master image was installed, one of the settings deactivated Samba. You must undo this change, by modifying the Bastille configuration file, `/etc/Bastille/config` and running **`bastille`** again. Edit the file with **`vi`**, search for the string `Samba` with the subcommand **`/Samba`** and set the value of `MiscellaneousDaemons.remotefs` from `Y` to **`N`**, as in Example 10-9:

Example 10-9 Searching for Samba in Bastille

```
# cd /etc/Bastille/
# cp config config.orig
# vi config   // Search for Samba (/Samba) and change Y to N
# Q:  Would you like to deactivate NFS and Samba? [Y]
MiscellaneousDaemons.remotefs="N"
```

Then run **`bastille`** with the **`-b`** flag which instructs it to get answers from the configuration file (Example 10-10).

Example 10-10 Running Bastille

```
# bastille -b
NOTE:    Entering Critical Code Execution.
         Bastille has disabled keyboard interrupts.
...
...
#########################################################
Errors have occurred in the configuration.
Please view the following file for more details:
        /var/log/Bastille/error-log
#########################################################
```

Do not be concerned with the last error message. It is related to the file `/etc/grub.conf`, which does not exist on zSeries Linux.

10.3.4 Configuring 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 good test directory is `/usr/share/doc/` as it has much good Linux documentation. TExample 10-11 creates a file share named `sharedoc`:

Example 10-11 Configuring smb.conf

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf    // add three lines at the bottom of the file:
...
[sharedoc]
        comment = SLES9 on zSeries documentation
        path = /usr/share/doc/
```

This causes an SMB share named **`sharedoc`** consisting of the contents of `/usr/share/doc` to be created when Samba is started.

10.3.5 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`, etc). To create a new Samba user, the **`smbpasswd -a`** command is used. Example 10-12 shows adding the user `mikem` first to Linux then to the `smbpasswd` file.

Example 10-12 Adding the mikem user ID

```
# id mikem
id: mikem: No such user
# useradd mikem
# passwd mikem
Changing password for mikem.
New password:
```

```
Re-enter new password:
Password changed
# mkdir /home/mikem
# chown mikem.users /home/mikem
# smbpasswd -a mikem
New SMB password:
Retype new SMB password:
Added user mikem.
```

You can see that the last **smbpasswd** command added `mikem` to the file `smbpasswd`:

```
# cat smbpasswd
...
mikem:1001:F3265269D0AC8A0E944E2DF489A880E4:DF43568E4C68049E43A6B09EBB041A6:[U]:LCT-41F8
EB94:
```

This method of maintaining Samba users, groups and passwords is good for a small number of users. For a larger number of users, merging Samba and LDAP is recommended. It is not as simple as pointing the virtual file and print server at the virtual LDAP server as described in 10.2.5, “Setting another virtual server to use LDAP server” on page 154, because the Samba schema must first be added to LDAP. Details are outside the scope of this book, but there are related presentations, *Directory Serving Solutions Using OpenLDAP* and *File Serving Solutions Using Samba-3* on the Web:

`http://linuxvm.org/present/`

10.3.6 Starting Samba at boot time

Samba consists of two daemons **nmbd** and **smbd**. They can be started for the current session with the **rcnmbd** and **rcsmbd** commands (Example 10-13).

Example 10-13 Starting the Samba deamons

```
# rcnmb start
Starting Samba NMB daemon
done
# rcsmb 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.

10.3.7 Testing your changes

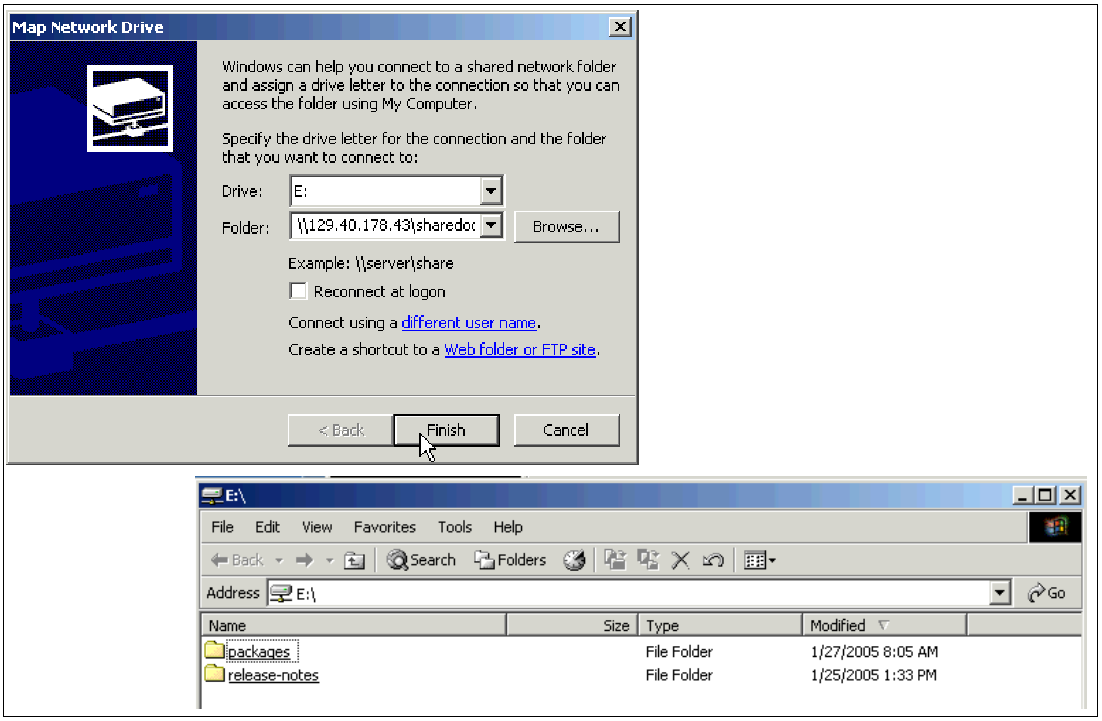
You can verify that the Samba daemons are running via the **status** parameter to the **rcsmb** and **rcnmb** commands (Example 10-14).

Example 10-14 Verifying the running daemons

```
# rcnmb status
Checking for Samba NMB daemon
running
# rcsmb status
```

```
Checking for Samba SMB daemon
running
```

You can test getting a Samba share from a Windows desktop. Go to any Windows Explorer window (such as My Computer) and select **Tools** → **Map Network Drive**. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 10-9. In this example the UNC is \\129.40.178.133\sharedoc. Then 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 the figure.



Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: E:

Folder: \\129.40.178.43\sharedoc

Example: \\server\share

☐ Reconnect at logon

Connect using a different user name.

Create a shortcut to a Web folder or FTP site.

Address: E:\

Name	Size	Type	Modified
packages		File Folder	1/27/2005 8:05 AM
release-notes		File Folder	1/25/2005 1:33 PM

Figure 10-9 Mapping a network drive to the Samba server

You should now have Samba configured and running with one new share available.

10.3.8 Configuring printing

Configuring printing is more complex and is beyond the scope of this section. For many more details see the Redpaper *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3863.

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 are software available on Linux to help form a development system for developers to create integrated applications. MySQL and Apache are among them. 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`.

10.4.1 Cloning a Linux virtual server

From the controller clone a basic virtual server to `LINUX04`.

```
# clone.sh linux04
...
```

SSH in to the new virtual server.

10.4.2 Scripting Languages

Scripts are good for quickly automating a process or writing your own commands. They are also used for being the backbone of robust applications. There are numerous scripting languages used in Linux application development, here are overviews of the most popular and general ones, obtained from their package descriptions.

- *Python* is an interpreted, object-oriented programming language, and is often compared to Tcl, Perl, Scheme, or Java. You can find an overview of Python in the documentation and tutorials included in the python-doc (HTML) or python-doc-pdf (PDF) packages. To install the python interpreter, execute the command:

  ```
  # yast -i python
  ```

- *Practical Extraction and Report Language (Perl).* Perl is optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is also good for many system management tasks. Perl is intended to be practical (easy to use, efficient, and complete) rather than beautiful (tiny, elegant, and minimal). To install perl, execute the command:

  ```
  # yast -i perl
  ```

- The *Tool Command Language* (Tcl), is a very simple programming language. Tcl provides basic language features such as variables, procedures, and control. It runs on almost any modern OS, such as Unix, Macintosh, and Windows 95/98/NT computers. The key feature of Tcl is its extensibility. Tcl was originally developed as a reusable command language for experimental Computer Aided Design (CAD) tools. The interpreter was implemented as a C library which could be linked to any application. It is very easy to add new functionality to the Tcl interpreter, so it is an ideal, reusable “macro language” that can be integrated into many applications. One of Tcl's best loved features is the ease with which one can add new commands (known as extensions). New commands can range from something as simple as a new format for producing output, to extensions such as Tk which provide graphically oriented programming paradigms. Another very popular extension is Expect which can be used to automate console-based interactive applications. To install tcl, execute the command

  ```
  # yast -i tcl
  ```

- *Hypertext Preprocessor* (PHP) is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. PHP development is focused on server-side scripting, but you can do much more with it. 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's most frequently used to compile C and C++ code. In the minimal SLES9 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
gcc-3.3.3-43.24
```

gcc does preprocessing, compilation, assembly and linking for files with extensions `.c`, `.cpp`, and 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 (that has been successfully compiled) to see where it is failing. Install it using yast:

```
# yast -i gdb
```

There is a good tutorial on getting 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 on 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

SLES9 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. Make sure you have the right Java package, if not or you are not sure which one you need, just install the SDK:

```
# yast -i IBMJava2-SDK
```

Again, you should do 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`. Here's an 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 aren't 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 just need one `.jar` file, instead of numerous class files.

1. First create a manifest file that specifies where the main method is by adding a simple one-liner. You must specify the package name as well:

   ```
   $ vi mainClass
   Main-Class: myHello.HelloWorld
   ```

2. Now use the **jar** command to create the executable jar file that knows where the main method is (Example 10-15). 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:

Example 10-15 Creating an executiable .jar file for the main method location

```
$ 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
```

3. Run the `.jar` file by invoking **java** with the **-jar** option.

   ```
   $ java -jar myHello.jar
   Hello World!
   ```

A good Java debugger is **jdb**, it comes with IBMJava2-SDK and can be run similar to **gdb**. A good tutorial is 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 the more recent and popular Ant. Ant uses XML technology. Here's a great guide 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.htmll
http://www.gnu.org/software/make/manual/html_chapter/make_toc.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

11

Miscellaneous recipes

This chapter has the following sections of miscellaneous tasks that you might want to perform:

11.1 Adding a logical volume

Often more disk space is needed than the 2 GB root file system used thus far. Two 3390-3s combined into a logical volume will give approximately 4.4GB of disk space. The logical volume manager (LVM) which is built into SLES9 can do this for you.

The overall steps in adding a logical volume are:

1. Add minidisks to the z/VM directory entry and IPL Linux
2. Install the LVM2 RPM
3. Bring the new DASD online
4. Format and partition the DASD
5. Create the logical volume and file system
6. Update the file system table
7. Make the change persistent

11.1.1 Adding minidisks to the z/VM directory entry

While we do not go into any specifics, the overall steps are on z/VM are:

1. Determine the labels of the two volumes that will be added.
2. Add minidisk statements to define minidisks at virtual addresses 103 and 104 to the appropriate Linux user ID definition in the `USER DIRECT` file.
3. Create the `USER DISKMAP` file to verify the disk layout.
4. Bring the changes online with the **`DIRECTXA`** command.
5. Logoff the Linux user ID and log back on to obtain the new directory entry. Run **`QUERY DASD`** to verify the new minidisks are available.
6. IPL Linux.

11.1.2 Installing LVM2 RPM

Start an ssh (PuTTY) session to the virtual server to which you added the minidisks.

The lvm2 RPM must be added. You could use the command **`yast -i lvm2`** to add it, but what if you do not know the name of the RPM you want to add? In this case, you can use **`yast`** to search for RPMs. Invoke **`yast`** (which can be done from a terminal session, yast2 requires a graphical session) and use the Tab and down-arrow keys to select **`Software`** → **`Install and Remove Software`**, as in Example 11-1.

Example 11-1 Opening a yast session

```
# yast

YaST @ pbc4553                                                            Press F1 for
Help

  +-----------------------------------------------------------------------------------------+
  ¦                                     YaST Control Center                                 ¦
  +-----------------------------------------------------------------------------------------+

  +--------------------------+ +------------------------------------------------------------+
  ¦Software                  ¦ ¦Online Update                                               ¦
  ¦Hardware                  ¦ ¦Install and Remove Software                                 ¦
  ¦System                    ¦ ¦Change Source of Installation                               ¦
  ¦Network Devices           ¦ ¦Installation into Directory                                 ¦
  ¦Network Services          ¦ ¦Patch CD Update                                             ¦
  ¦Security and Users        ¦ ¦System Update                                               ¦
  ¦Misc                     ¦ ¦YOU Server Configuration                                ¦
...
```

This should bring you to a panel with some RPMs listed. To search for RPMs, use the Tab key to move to **Search** at the bottom. This should bring you to a Package Search panel. Type in **`lvm`** for a search string, move the cursor to the **OK** button and press **Enter**.

Example 11-2 Searching on lvm in Package Search

```
                  Package Search

+-------------------------------------------------+
¦Search Phrase                                    ¦
¦lvm ¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦-¦
+-------------------------------------------------+
                 [x] Ignore Case

+ Search in --------------------------------------+
¦[x] Name of the Package                          ¦
¦[x] Summary                                      ¦
¦[ ] Description (time-consuming)                 ¦
¦[ ] Provides                                     ¦
¦[ ] Requires                                     ¦
+-------------------------------------------------+

       [   OK    ]       [   Cancel    ]
```

This search should result in one RPM named `lvm2`. Press the Space bar to select the package and a “+” character should be displayed to the left of the package.

```
¦     ¦Name¦Avail. Vers.¦Inst. Vers.¦Summary   ¦Size       ¦Source
¦  +  ¦lvm2¦2.00.33     ¦           ¦LVM2 tools¦   638.9 kB¦
```

Click **Accept**. The RPM should be added. **Quit** yast. You can verify the RPM was added with the **`rpm`** command.

11.1.3 Bringing the new minidisks online

The **`lsdasd`** command will show the minidisks that are active. Note that the new 103 and 104 DASD are not shown:

```
# lsdasd
0.0.0100(ECKD) at ( 94: 0) is dasda : active at blocksize 4096, 360000 blocks, 1406 MB
0.0.0101(ECKD) at ( 94: 4) is dasdb : active at blocksize 4096, 36000 blocks, 140 MB
```

The **`lscss`** command lists the channel subsystem devices. The **`-t 3390`** flag shows only the DASD devices. Note that in Example 11-3, 103 and 104 are listed but not in use.

Example 11-3 Using lscss with -t 3390

```
# lscss -t 3390
Device   Subchan.  DevType CU Type Use  PIM PAM POM  CHPIDs
----------------------------------------------------------------------
0.0.0100 0.0.0000  3390/0A 3990/E9 yes  F0  F0  FF   8C8D8E8F 00000000
0.0.0101 0.0.0001  3390/0A 3990/E9 yes  F0  F0  FF   8C8D8E8F 00000000
0.0.0103 0.0.0002  3390/0A 3990/E9      F0  F0  FF   8C8D8E8F 00000000
0.0.0104 0.0.0003  3390/0A 3990/E9      F0  F0  FF   8C8D8E8F 00000000
0.0.0592 0.0.000E  3390/0A 3990/E9      F0  F0  FF   8C8D8E8F 00000000
```

```
...
```

The **chccwdev** command with the **-e** flag will activate the unused DASD. Use the **lsdasd** command again to verify they are online (Example 11-4).

Example 11-4 Using chccwdev -e

```
# chccwdev -e 0.0.0103
Setting device 0.0.0103 online
Done
# chccwdev -e 0.0.0104
Setting device 0.0.0104 online
Done
# lsdasd
0.0.0100(ECKD) at ( 94:  0) is dasda : active at blocksize 4096, 360000 blocks, 1406 MB
0.0.0101(ECKD) at ( 94:  4) is dasdb : active at blocksize 4096, 36000 blocks, 140 MB
0.0.0103(ECKD) at ( 94:  8) is dasdc : active at blocksize 4096, 204840 blocks, 800 MB
0.0.0104(ECKD) at ( 94: 12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
```

11.1.4 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 so as to format both minidisks in parallel (Example 11-5).

Example 11-5 Using the for loop

```
# for i in c d
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 2713
[2] 2714
```

When the jobs are finished use the **fdasd** command with the **-a** flag to create a single partition from each minidisk (Example 11-6).

Example 11-6 Using fdasd -a

```
# fdasd -a /dev/dasdc
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
# fdasd -a /dev/dasdd
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The minidisks should now be ready to use to create a logical volume

11.1.5 Create the logical volume and file system

The overall steps involved in creating a logical volume are as follow:

1. Create physical volumes from the two DASD.
2. Create a single volume group.

3. Create a single logical volume.
4. Make a file system from the logical volume.

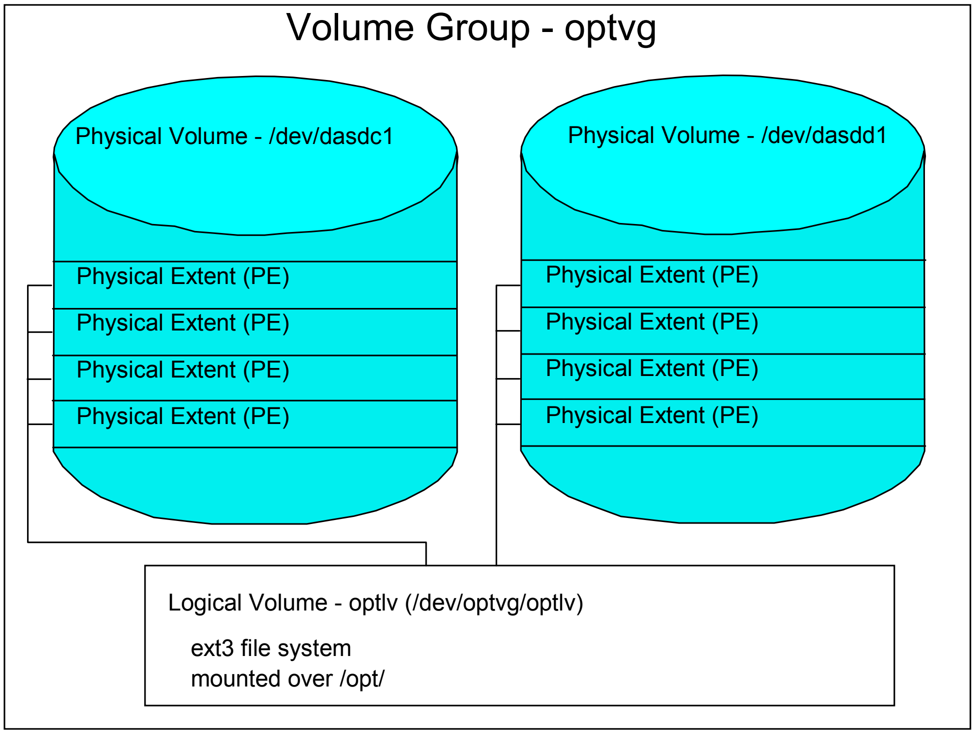


Figure 11-1 LVM block diagram

Figure 11-1 shows a block diagram of the logical volume manager reflecting this example.

Creating physical volumes from the two DASD

In Example 11-7, the **pvcreate** command initializes DASD for use by LVM. Initialize the two new DASD partitions. Verify with the **pvdisplay** command.

Example 11-7 Using pvcreate and pvdisplay

```
# pvcreate /dev/dasdc1 /dev/dasdd1
  Physical volume "/dev/dasdc1" successfully created
  Physical volume "/dev/dasdd1" successfully created
# pvdisplay
  --- NEW Physical volume ---
  PV Name               /dev/dasdc1
  VG Name
  PV Size               799.88 MB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               82c0rk-iVg8-jvs8-6CLg-qjtB-YaLD-wiG21w

  --- NEW Physical volume ---
  PV Name               /dev/dasdd1
  VG Name
  PV Size               2.29 GB
...
```

Creating a single volume group

In Example 11-8, the **`vgcreate`** command can be used to create a volume group named `optvg` from the two DASD. Use the **`vgdisplay`** command to verify.

Example 11-8 Using vgcreate and vgdisplay

```
# vgcreate optvg /dev/dasdc1 /dev/dasdd1
  Volume group "optvg" successfully created
# vgdisplay
  --- Volume group ---
  VG Name               optvg
  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               3.07 GB
  PE Size               4.00 MB
  Total PE              785
  Alloc PE / Size       0 / 0
  Free  PE / Size       785 / 3.07 GB
  VG UUID               aOSwJA-OBWk-M8f8-OgXY-eDUl-jb8T-Ug3Vo5
```

In this example, there are 785 free physical extents.

Creating a single logical volume

The **`lvcreate`** command is used to create a logical volume, as in Example 11-9. The **`-l 785`** flag specifies to use all free extents, in this example. The **`-n optlv`** specifies the name of the logical volume. The last argument **`optvg`** specifies the name of the volume group from which the logical volume will be created. Use the **`lvdisplay`** command to verify.

Example 11-9 Using lvcreate and lvdisplay

```
# lvcreate -l 785 -n optlv optvg
  Logical volume "optlv" created
# lvdisplay
  --- Logical volume ---
  LV Name                /dev/optvg/optlv
  VG Name                optvg
  LV UUID                GriEkv-peJd-C85c-d1xO-87IR-iVa4-Z79gT8
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                3.07 GB
  Current LE             785
  Segments               2
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0
```

Making a file system from the logical volume

Now you have a logical volume. As in Example 11-10, 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.

Example 11-10 Using mke2fs

```
# mke2fs -j /dev/optvg/optlv
mke2fs 1.36 (05-Feb-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
402400 inodes, 803840 blocks
40192 blocks (5.00%) reserved for the super user
...
```

The file system created from the logical volume is now ready to be mounted.

11.1.6 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. First make a backup copy of the `fstab` file (Example 11-11).

Example 11-11 Backing up fstab

```
# cd /etc
# cp fstab fstab.orig
# vi fstab
/dev/dasda1         /                    ext3      acl,user_xattr       1 1
/dev/dasdb1         swap                 swap      pri=42               0 0
/dev/optvg/optlv    /opt                 ext3      acl,user_xattr       0 0
...
```

Note that the `/opt/` file system is empty (if there were data in the directory you might have copied it out, emptied `/opt/` and then copied it back after the mount). Mount the `/opt/` file system with one argument. Use the **`df -h`** command to verify that it is mounted, as in Example 11-12.

Example 11-12 Mounting the opt file

```
# ls /opt
.  ..
# mount /opt
# ls /opt
.  ..  lost+found
# df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/dasda1          1.4G  515M  800M  40% /
tmpfs                378M     0  378M   0% /dev/shm
/dev/mapper/optvg-optlv
                     4.4G   33M  4.3G   2% /opt
```

11.1.7 Making the change persistent

The new logical volume is mounted over `/opt/`. However, will the change be persistent across reboots? Even though the `fstab` file was updated, the answer is no because the system does not know to bring the new DASD online.

There are two ways to make this change effective. It is possible to leave the minidisk (DASD) virtual device addresses out of the zipl configuration file, `/etc/zipl.conf`, or they can be added to that file. It is recommended that you specifically include minidisk addresses in the zipl configuration file so you will be able to remember which minidisks are part of the system. This is done by adding the `dasd=` parameter to the boot parameters. When specifying it you must maintain the ordering of the existing minidisks (`/dev/dasda` = 100, `/dev/dasdb` = 101 and `/dev/dasdc` = 102), then add the new minidisks (103 and 104). You could specify the string `dasd=100-102,103-104`, or that can be shortened to `dasd=100-104` because the addresses are consecutive. If you chose different devices addresses for the new minidisks, such as 200 and 201, the string would be `dasd=100-102,200-201`.

Make a backup copy of `zipl.conf` then modify the original by adding the parameter string `dasd=100-104, as in` Example 11-13.

Example 11-13 Backing up zip1.conf

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf
# Modified by YaST2. Last modification on Wed Nov 30 15:16:43 2005

[defaultboot]
    default = ipl

[ipl]
    target = /boot/zipl
    image = /boot/image
    ramdisk = /boot/initrd,0x1000000
    parameters = "root=/dev/dasda1 dasd=100-104 selinux=0 TERM=dumb elevator=cfq"
...
```

Now run the **`mkinitrd`** and **`zipl`** commands to bring the changes online (Example 11-14).

Example 11-14 Using mkinitrd and zip1 to effect the changes

```
# mkinitrd
...
# mkinitrd
Root device:    /dev/dasda1 (mounted on / as ext3)
Module list:    jbd ext3 dasd_eckd_mod dasd_fba_mod

Kernel image:   /boot/image-2.6.5-7.244-s390x
Initrd image:   /boot/initrd-2.6.5-7.244-s390x
Shared libs:    lib64/ld-2.3.3.so lib64/libblkid.so.1.0 lib64/libc.so.6
lib64/libselinux.so.1 lib64/libuuid.so.1.2
Modules:        kernel/fs/jbd/jbd.ko kernel/fs/ext3/ext3.ko
kernel/drivers/s390/block/dasd_mod.ko kernel/drivers/s390/block/dasd_eckd_mod.ko
kernel/drivers/s390/block/dasd_fba_mod.ko
DASDs:         0.0.0100(ECKD) 0.0.0101(ECKD) 0.0.0102(ECKD) 0.0.0103(ECKD) 0.0.0104(ECKD)
Including:      udev

initrd updated, zipl needs to update the IPL record before IPL!
# zipl
Using config file '/etc/zipl.conf'
```

```
Building bootmap in '/boot/zipl'
Adding IPL section 'ipl' (default)
Preparing boot device: dasda (0100).
Done.
```

You can now **`reboot`** the system to test the changes (Example 11-15). If you are quick, you can **`exit`** the SSH session before you lose communication.

Example 11-15 Rebooting the system

```
# reboot

Broadcast message from root (pts/0) (Wed Nov 30 13:47:14 2005):

The system is going down for reboot NOW!
# exit
```

When your system comes back the logical volume should be mounted over the /opt/ directory.

11.2 Creating a DCSS/XIP2 shared file system

This section contains excerpts from the document *How to use Execute-in-Place Technology with Linux on z/VM*, SC33-8283-00, March 23, 2005. A copy of this document can be found in:

`http://awlinux1.alphaworks.ibm.com/developerworks/linux390/docu/l26bhe00.pdf`

This section describes how you can reduce memory requirements of Linux servers by using z/VM Discontiguous Saved Segments (DCSS). All operating system instances that run concurrently on a mainframe vie for some of the available physical memory. When multiple operating system instances need the same data it often gets loaded into memory several times. Linux itself only loads shared libraries into memory once, but DCSS would optimize that further by allowing you to customize shared binaries and libraries and by having only one copy loaded in memory per VM instead of per guest.

A major part of the memory required by a Linux server is used for binary application files and for shared library files. The shared data must be identical across all sharing Linux instances. Some suitable candidates are:

- Applications files can be shared, only by Linux instances that use the same version of the application.
- Make shared data read-only. This is to prevent Linux instances from interfering with one another.
- Directories with applications and libraries that are frequently used by numerous Linux instances are good candidates for sharing.

Directories that are *not* suitable for sharing are:

- Directories that are written to.
- Script-sharing is not effective because it is interpreted and not executed directly. This is not to be confused with the interpreters themselves (bash, perl), which can be shared.

For a detailed description of how sharing works, see the document referenced at the start of this section.

11.2.1 Creating a DCSS

You first need to determine the maximum size of the DCSS.

1. Determine the beginning memory address of the DCSS address range. The address range must not overlap with the virtual storage of the guest operating system. Therefore the beginning memory address of the DCSS is equal to the highest amount of virtual storage that has been allocated to any of your images.
2. Determine the upper boundary of the DCSS address range. The upper boundary is 1960 MB for 31-bit Linux kernels and 2 GB for 64-bit Linux kernels.
3. Subtract the lower beginning memory address from the upper boundary to obtain the maximum possible DCSS size.

The maximum possible size of the DCSS in this example assumes a 31-bit Linux distribution and 256 MB of virtual storage:

```
1960 MB - 256 MB = 1704 MB
```

Identifying directories to be shared

Good candidates are those which are read-only files/directories that are at the same level (version) in all Linux instances and that are applications and libraries that are frequently used by most of the images. Sharing the following directories can be used to test setting up a DCSS. To choose directories to share that results in performance improvements, you will have to profile a test workload that your Linux systems will be running.

- `/lib/`
- `/usr/lib/`
- `/bin/`
- `/usr/bin/`
- `/sbin/`
- `/usr/sbin/`

To calculate the space requirements for sharing the directories above, use the following command in Example 11-16 to find the space requirements for the chosen directories:

Example 11-16 Using du -smc

```
# du -smc /lib /usr/lib /bin /usr/bin /sbin /usr/sbin
31      /lib
180     /usr/lib
8       /bin
22      /usr/bin
12      /sbin
7       /usr/sbin
257     total
```

The space used by the directories adds up to 257 MB. You should add about 10% for metadata and contingencies which results in 282 MB. Rounding up to a multiple of 64 MB gives 320 MB which is well below the maximum size of 1704 MB.

To establish the start and end address for the DCSS, this addresses must be on a page boundary (4 KB on the mainframe) and in hexadecimal notation.

Start address: 256 MB => `0x10000000`

End address: 256 MB + 320 MB - 1 B => `0x23ffffff`

Calculate the page frame number for the start and end address. You can do this by dividing the above numbers by 4096 (4K pages). In hexadecimal notation 4096 = 1000, so dividing the above numbers by 1000 can be accomplished by dropping the last three digits.

Start address: `0x10000000` = page frame no. `0x10000`

End address: `0x23ffffff` = page frame no. `0x23fff`

11.2.2 Preparing the Linux guest for DCSS creation

Logon to `MAINT` and edit the user directory. Change the user ID that will be used to create the DCSS. In this example it is `LINUX01`. Add privilege class E to the user definition which is needed to define and load a new DCSS. If not set already, then set the maximum memory allocation to 1G, to allow enough memory for loading the DCSS:

```
==> x user direct
```

The `LINUX01` user definition should look like this:

```
user linux01 lnx4vm 256M 1G eg
```

Put the directory online after you file the changes.

```
====> file
==> directxa user
```

11.2.3 Creating a file system image for the DCSS

A DCSS holds shared Linux code in form of a Linux file system. Before you can save a file system as a DCSS, you need to IPL from a disk that contains the file system image. So the first thing you need to do is create the file system image on a separate disk. This disk is needed temporarily only, for initializing the DCSS. You can use a temporary disk, a minidisk or a virtual disk. In this example a virtual disk or VDISK is used.

1. Shut down the Linux image running on guest `LINUX01` and logon to `LINUX01` in a 3270 emulator session.
2. A virtual disk can be used to house the DCSS file system. This disk is only temporary, once you define the shared segment you will not need this anymore. You can create a VDISK with the following **`DEFINE`** command:

   ```
   ==> def vfb-512 as 103 blk 1048576
   00: DASD 0103 DEFINED
   ```

   This VDISK is 1048576 blocks, or 512MB, which is big enough to house the directories to be shared (see “Identifying directories to be shared” on page 176).
3. IPL the Linux image:

   ```
   ==> ipl 100 clear
   ```

4. Open a SSH session to the Linux image and login as root.
5. Invoke **`yast`** to activate the DASD (you could also use **`chccwdev -e`**).

   ```
   # yast
   ```

6. Select **Hardware** → **DASD** → Select **0.0.0103** → **Perform Action** → **Activate.**
7. Find out which device is the VDISK defined in step 1 (Example 11-17).

Example 11-17 Determining the VDISK

```
# cat /proc/dasd/devices
0.0.0100(ECKD) at ( 94: 0) is dasda : active at blocksize: 4096, 546840 blocks, 2136 MB
0.0.0101(FBA ) at ( 94: 4) is dasdb : active at blocksize: 512, 524288 blocks, 256 MB
```

```
0.0.0102(ECKD) at ( 94: 8) is dasdc : active at blocksize: 4096, 54000 blocks, 210 MB
0.0.0103(FBA ) at ( 94: 12) is dasdd : active at blocksize: 512, 1048576 blocks, 512 MB
```

Note that the 103 minidisk is `dasdd`.

8. Run the **`fdisk`** command to create a partition for the FBA device, as in Example 11-18. Use the **`p`** subcommand to see that there are no existing partitions.

Example 11-18 Usng fdisk to create a partition

```
# fdisk /dev/dasdd
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): p

Disk /dev/dasdd: 447 MB, 447393792 bytes
16 heads, 128 sectors/track, 426 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

     Device Boot      Start         End      Blocks   Id  System
```

9. Create a new primary partition using all of the space with the **`n`** subcommand, as in Example 11-19.

Example 11-19 Creating a new primary partition

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-426, default 1): Enter
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-426, default 426): Enter
Using default value 426
```

10. Save your changes with the **`w`** subcommand, as in Example 11-20. The new partition will be `/dev/dasdd1`.

Example 11-20 Saving changes

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

11. Create a file system in the partition with the **`mke2fs`** command (Example 11-21).

Example 11-21 Creating a file system

```
# mke2fs /dev/dasdd1
mke2fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
```

```
Fragment size=1024 (log=0)
131072 inodes, 524224 blocks
26211 blocks (5.00%) reserved for the super user
First data block=1
64 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

12. Create an empty mount point, `/dcss/` and mount the new file system over it, as in Example 11-22.

Example 11-22 Creaging an empty mount point and mounting the file system

```
# mkdir /dcss
# mount /dev/dasdd1 /dcss
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasda1            2152876    997204   1046312  49% /
tmpfs                   117320         4    117316   1% /dev/shm
/dev/dasdd1             507684        13    481460   1% /dcss
```

13. Create an empty file with the size of the DCSS with the **dd** command. The total number of 4K pages needed to hold a 320 MB file system is `320M/4K = 81920` (Example 11-23).

Example 11-23 Creating an emply file

```
# dd if=/dev/zero  of=/dcss/filesystem   bs=4096 count=81920
81920+0 records in
81920+0 records out
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.1G  974M 1022M  49% /
tmpfs                 115M  4.0K  115M   1% /dev/shm
/dev/dasdd1           496M  322M  149M  69% /dcss
```

14. Create an empty ext2 file system in the empty file using the **mke2fs** command, asin Example 11-24. You are informed that `filesystem` is not a block device and asked if you really want to create a file system in it. Respond with **y**.

Example 11-24 Creaing an emply ext2 file system

```
# mke2fs -b 4096   /dcss/filesystem
mke2fs 1.38 (30-Jun-2005)
/dcss/filesystem is not a block special device.
Proceed anyway? (y,n) y
...
This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

15. Create a mount point, `/xipimage/` for the file system.

```
# mkdir /xipimage
```

16. Mount the file system on the mount point you have created. The **`-t ext2`** specifies the file system type and the **`-o loop`** specifies this is a loopback mount:

```
# mount -t ext2 -o loop /dcss/filesystem /xipimage
```

17. Double check that you have a copy of the excute-in-place script, **`xipint`**, in `/sbin/`. The **`xipinit`** script should have been copied to the master image before cloning - it is part of the package that comes with this book. The source code is in Appendix A.6.3, “The xipinit script” on page 238.
18. Copy all directories to be shared to the file system image, as in Example 11-25.

Example 11-25 copying all shared directories to the file system

```
# cp -ap /bin /xipimage/bin
# cp -ap /lib /xipimage/lib
# cp -ap /sbin /xipimage/sbin
# mkdir /xipimage/usr
# cp -ap /usr/bin /xipimage/usr/bin
# cp -ap /usr/lib /xipimage/usr/lib
# cp -ap /usr/sbin /xipimage/usr/sbin
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.1G  974M 1022M  49% /
tmpfs                 115M  4.0K  115M   1% /dev/shm
/dev/dasdd1           496M  322M  149M  69% /dcss
/dcss/filesystem      310M  267M   28M  91% /xipimage
```

19. Unmount the file system.

```
 # umount /xipimage/
```

Creating a DCSS from the image file

Use these steps to create the DCSS.

1. Before creating the DCSS in CMS, prepare the DASD with the image file for IPL, as in Example 11-26. Use the **`zipl`** command where the **`-t`** flag identifies the directory that contains the image file. Option -s identifies the image file and start address of the DCSS.

Example 11-26 Preparing the DASD

```
# zipl -t /dcss -s /dcss/filesystem,0x10000000
Building bootmap in '/dcss'
Adding segment load section
  segment file......: /dcss/filesystem at 0x10000000
Preparing boot device: dasdd (0103).
Done.
```

2. Shutdown the Linux system

```
# shutdown -h now
...
```

3. From a 3270 emulator session, IPL CMS but do not IPL Linux, as in Example 11-27.

Example 11-27 IPL CMAS only

```
==> i cms
z/VM V5.2.0    2006-01-24 13:26
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
Do you want to IPL Linux from DASD 100? y/n
```

```
==> n
```

Your Linux guest machine requires privilege class E to be able to add a DCSS, which was added previously.

4. Use the **QUERY NSS MAP** command. You should see that the named saved segment `BC05DCSS` has an **S** under the class column. S stands for skeleton. This means that it is just a placeholder for a DCSS but there are none yet saved.

```
==> q nss map
```

5. Before loading the DCSS, a segment must be defined. This is done by entering the command:

```
==> defseg bc05dcss 10000-23fff sr
00: HCPNSD440I Saved segment BC05DCSS was successfully defined in fileid 0029
```

6. In order to save the segment you just defined you cannot have more than one CPU available. If you have more than 1, you can temporarily detach the other CPU(s).

```
==> q cpus
00: CPU 00  ID  FF14ABAA20848000 (BASE)
00: CPU 01  ID  FF14ABAA20848000
==> det CPU 01
```

7. You want to be sure you have enough storage defined for the mapping of the DCSS. Remember that the start address is 256 MB and the DCSS is 320 MB. This means at least a 576 MB machine. Define storage to 1 GB just to make it simpler.

```
==> def stor 1g
00: STORAGE=1G
00: Storage cleared - system reset.
```

8. IPL the DASD that contains the file system to be copied into the new DCSS. The boot loader installed on the disk loads the contents of the DCSS into the z/VM virtual guest memory and then enters a disable wait state.

```
==> ipl 103 clear
00: Booting default...
00: HCPGIR450W CP entered; disabled wait PSW 000A0000 00000000
```

9. After the memory is loaded, save the DCSS via the **SAVESEG** command. This might take a couple of minutes:

```
==> saveseg bc05dcss
00: HCPNSS440I Saved segment BC05DCSS was successfully saved in fileid 0029.
```

10. Following the **saveseg** command, issue the command **QUERY NSS MAP** to verify that the new DCSS save operation was successful. It completed successfully if the class column (column title CL) has changed from skeleton (S) to active (A).

You have now created a DCSS in z/VM.

11.2.4 Configuring Linux to use the DCSS

Before your Linux kernel can use the DCSS, it must be aware of the extended address space that covers the DCSS.

Perform the following steps to set the **mem=** kernel parameter accordingly:

1. Reset the storage for your linux system to 256 MB. And if you had to detach CPUs for the **saveseg** command, now is the time the add them back.

```
==> def stor 256m
==> define cpu 01
```

2. IPL your Linux system.

```
==>  ipl 100 clear
```

3. Add a **mem=<value>** parameter to your kernel parameter line of the `zipl.conf` file. The value must cover the entire DCSS, that is, must be equal to or above the DCSS end address. The value can be in either of these forms: in byte, in the form <x>k, in the form <y>M. TExample 11-28 uses **mem=1024M**.

Example 11-28 Adding mem-1024M to zip1.conf

```
# vi /etc/zipl.conf
# Modified by YaST2. Last modification on Mon Dec 13 19:49:11 2004
[defaultboot]
    default = ipl
[ipl]
    target = /boot/zipl
    image = /boot/image
    ramdisk = /boot/initrd
    parameters = "root=/dev/dasda1 selinux=0 TERM=dumb elevator=cfq mem=1024M"
[dumpdasd]
    target = /boot/zipl
    dumpto = /dev/dasd??
[dumptape]
    target = /boot/zipl
    dumpto = /dev/rtibm0
```

4. Run **zipl** with the new parameter file.

```
# zipl
```

5. Restart Linux.

```
# shutdown –r now
```

6. Issue the following command to verify that Linux is using the new parameter:

```
# cat /proc/cmdline
root=/dev/dasda1 selinux=0 TERM=dumb elevator=cfq mem=1024M BOOT_IMAGE=0
```

11.2.5 Testing the DCSS

Perform the following steps to test the DCSS.

1. Mount the xip2 file system, as in Example 11-29.

Example 11-29 Mounting the xip2 file system

```
# mount -t xip2 -o ro,memarea=BC05DCSS none /dcss
extmem info:segment_load: loaded segment BC05DCSS range 0000000010000000 .. 0000
000023ffffff type SR in shared mode
Jan 12 10:46:52 lat131 kernel: extmem info:segment_load: loaded segment BC05DCSS
 range 0000000010000000 .. 0000000023ffffff type SR in shared mode
```

2. As in Example 11-30, make sure that all files are accessible.

Example 11-30 Accessing the files

```
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasda1            2152876    883988   1159528  44% /
tmpfs                   117316         0    117316   0% /dev/shm
none                    317392    250368     50640  84% /dcss
# cd /dcss
# ls -l
```

```
total 44
drwxr-xr-x   7 root root  4096 Dec 13 15:58 ./
drwxr-xr-x  21 root root  4096 Dec 14 11:33 ../
drwxr-xr-x   2 root root  4096 Dec 13 14:45 bin/
drwxr-xr-x  11 root root  4096 Dec 13 14:44 lib/
drwx------   2 root root 16384 Dec 13 15:54 lost+found/
drwxr-xr-x   3 root root  8192 Dec 13 14:46 sbin/
drwxr-xr-x   4 root root  4096 Dec 13 15:59 usr/
# ls -l usr
total 24
drwxr-xr-x   5 root root 4096 Jan 16 17:29 .
drwxr-xr-x   7 root root 4096 Jan 16 17:29 ..
drwxr-xr-x   2 root root 8192 Dec 23 14:56 bin
drwxr-xr-x  22 root root 4096 Dec 23 14:56 lib
drwxr-xr-x   2 root root 4096 Dec 23 14:56 sbin
```

11.2.6 Activating the execute-in-place file system at boot time

Before you can activate the execute-in-place file system you have to *over-mount* the shared directories on startup. A sample script, **`/sbin/xipinit`** has been provided that can be used to over-mount directories with the content of a DCSS before Linux accesses them. Over-mounting a directory means replacing it with the contents of the DCSS at system startup. Shared directories must be over-mounted before any data on them is accessed. If a directory is accessed before being over-mounted, accessed libraries and binaries might remain in and waste memory. Directories that are on the root file system need to be over-mounted before running the first program on system startup, `/sbin/init`.

On system startup, **`xipinit`** runs first, over-mounts the shared directories and then calls **`/sbin/init`**.

1. The **`xipinit`** script can be modified to change the directories that are to be shared. In Example 11-31, no modification to the file is necessary because the `RODIRS` variable is set correctly.

Example 11-31 Using xipinit

```
##########################################################################
#########  Change RODIRS to add the directories you are sharing.  ########
#########  Make sure you end RODIRS in a comma.                   ########
#########  Change ROMOUNT to the mount point of your choice.      ########
#########  Change DCSSNAME for the name of your DCSS.             ########
##########################################################################
RODIRS=/lib,/usr/lib,/bin,/usr/bin,/sbin,/usr/sbin,
ROMOUNT=/dcss
DCSSNAME=BCO5DCSS
...
```

2. Test the **`xipinit`** script by executing it and verifying that the directories are over-mounted correctly, asin Example 11-32.

Example 11-32 Testing xipinit

```
# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/dasda1           2152876    997236   1046280  49% /
tmpfs                  117320         4    117316   1% /dev/shm
# xipinit
binding directory /lib
binding directory /usr/lib
binding directory /bin
binding directory /usr/bin
binding directory /sbin
binding directory /usr/sbin
Usage: init 0123456SsQqAaBbCcUu

# cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
none /dcss xip2 ro 0 0
none /dcss xip2 ro 0 0
none /lib xip2 ro 0 0
none /usr/lib xip2 ro 0 0
none /bin xip2 ro 0 0
none /usr/bin xip2 ro 0 0
none /sbin xip2 ro 0 0
none /usr/sbin xip2 ro 0 0
```

3. As in Example 11-33, verify that the directories are mounted.

Example 11-33 Verfying the mounted directories

```
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasda1            2152876    997236   1046280  49% /
tmpfs                   117320         4    117316   1% /dev/shm
none                    317392    272476     28532  91% /dcss
/dcss/lib               317392    272476     28532  91% /lib
/dcss/usr/lib           317392    272476     28532  91% /usr/lib
/dcss/bin               317392    272476     28532  91% /bin
/dcss/usr/bin           317392    272476     28532  91% /usr/bin
/dcss/sbin              317392    272476     28532  91% /sbin
/dcss/usr/sbin          317392    272476     28532  91% /usr/sbin
```

4. Now that the directories are correctly mounted, change the kernel parameters so that **xipinit** is run at boot time. Edit the kernel parameter file and add **init=/sbin/xipinit** to the kernel parameter line, as in Example 11-34.

Example 11-34 Changing kernel paramaters

```
# vi /etc/zipl.conf
# Modified by YaST2. Last modification on Mon Dec 13 19:49:11 2004

[defaultboot]
    default = ipl

[ipl]
    target = /boot/zipl
    image = /boot/image
    ramdisk = /boot/initrd
    parameters = "root=/dev/dasda1 selinux=0 TERM=dumb elevator=cfq mem=1024M init=/sbin/xipinit"

[dumpdasd]
    target = /boot/zipl
    dumpto = /dev/dasd??
[dumptape]
    target = /boot/zipl
```

```
dumpto = /dev/rtibm0
```

5. Run **`zipl`** with the new parameter file.

   ```
   # zipl
   ```

6. Reboot Linux.

   ```
   # shutdown -r now
   ```

7. When your system comes back up, verify all the shared directories are over-mounted (Example 11-35).

Example 11-35 Verfying overmounted directories

```
# cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw 0 0
none /dcss xip2 ro 0 0
none /lib xip2 ro 0 0
none /usr/lib xip2 ro 0 0
none /bin xip2 ro 0 0
none /usr/bin xip2 ro 0 0
none /sbin xip2 ro 0 0
none /usr/sbin xip2 ro 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
```

8. From a 3270 session verify that Linux is using the DCSS by issuing the **`QUERY NSS MAP`** command. Check the #USERS column; it should be equal to the number of users that successfully connected to the DCSS.

   ```
   ==> #cp q nss map
   FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
   ...
   0037 BC05DCSS DCSS       N/A    20000  3BFFF   SR  A  00001   N/A       N/A
   ```

11.2.7 Modifying another Linux guest to mount DCSS

Now change another Linux system, such as `LINUX02`, to mount the DCSS. Here is a high-level recipe of these steps, without the details.

1. Add the **`mem=1024M`** parameter to `zipl.conf.`
2. Run **`zipl`** and reboot.
3. Copy **`xipinit`** to `/sbin/.`
4. Make the directory `/dcss/.`
5. Test **`xipinit.`**
6. Modify `zipl.conf` to add **`xipinit.`**
7. Run **`zipl`** and reboot.

12

Monitoring z/VM and Linux

This chapter briefly describes how to monitor z/VM and Linux. For a more thorough chapter on z/VM performance and monitoring, see Chapter 11, *Monitoring performance and capacity*, in the Manual *Getting Started With Linux*, SC24-6096 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 Computer Associates' VM:Monitor, IBM z/VM Performance Toolkit and Velocity Software's ESALPS. The IBM z/VM Performance Toolkit is briefly described in this section.

For more information about Computer Associates' VM:Monitor, see:

`http://www.ca.com/`

For more information about Velocity's ESALPS, see:

`http://www.velocitysoftware.com/`

This chapter describes the following topics:

- 12.1, "Using INDICATE and other commands" on page 188
- 12.2, "The z/VM Performance Toolkit" on page 191
- 12.3, "Monitoring Linux" on page 197
- 12.4, "Registering Linux images with the Performance Toolkit" on page 200

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 we address here.

12.1.1 Using the INDICATE command

z/VM has some basic commands such as **`INDICATE`**. There are many parameters that can be included. Use the command **`HELP INDICATE`** for a basic understanding and then press F11 for help on each parameter.

INIDICATE LOAD

If no parameter is specified **`INDICATE LOAD`** is the default option. There are two flavors of this, a class G and a class E. Class G users can use **`INDICATE`** to display recent contention for system resources, display environment characteristics and measurements of resources used by their virtual machine.

The output from user ID with class E privilege (e.g. `MAINT`, `OPERATOR`) is shown here. The lines in Example 12-1 are number for clarity in the following description.

Example 12-1 INDICATE LOAD parameter

```
==> 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:

- `AVGPROC` on line 1 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. Take a glance at these to see if they are somewhat balanced. There are cases where an imbalance is okay. This would include very low utilization scenarios or cases where there are not enough ready to run virtual processors to keep the physical processors busy. Line 2 describes paging to expanded storage. Most z/VM systems on z900 class machines can sustain 1000s 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. However, there are times the `MIGRATE` value will spike for brief periods of time.
- Minidisk cache (`MDC`) statistics are given on the third line. 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. However, a high `HIT`

`RATIO` with a low value for the READS rate is not good (it doesn't matter much if you have a 100% hit ratio, but are doing only 1 I/O per second).

- Line **4** describes more storage (memory) management. The `PAGING` rate is important. Higher values will often impact performance. The `STEAL` percentage is often misleading. This is basically the percentage of pages taken from guests that z/VM believes are non-dormant. Since 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 considered stolen. So there are scenarios where a system only has 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 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 3 different classes (1 through 3) and a special class of zero. So the Column of Q_x values and E_x represent the virtual machines in the dispatch list and the eligible list. The most important value 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. Do not worry about the values in parenthesis.

INDICATE QUEUES EXP

Another useful command to understand the state of the system is the **`INDICATE QUEUES EXP, in`** Example 12-2.

Example 12-2 INDICATE QUEUES EXP

```
==> 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 displays the virtual processors (a single virtual machine can have multiple virtual processors) what queue (dispatch list, eligible, limit list) they are in and what state they are. This is a snapshot in time. Again you want to make sure there are not any 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 also gives state of virtual processor. This can be helpful to get a feel for how the virtual processors might be constrained. Virtual processors that are actually running at the snapshot are marked with and 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 not an available processor. (**Note**: the virtual machine that issues the **`INDICATE`** command will always be one of the running machines). Other states are documented in the help for **`IND Q EXP`**. You do not have to be concerned about the other columns unless detailed analysis is required or if IBM support requests it. Also, always remember that is just a snapshot in time so often repeating this command over time can give a more accurate picture of your z/VM system.

12.1.2 Using other basic commands

Some other useful basic commands are briefly mentioned in this section. 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. Sometimes it’s hard to find help for exactly the command you’re looking for. Some useful help commands are in Example 12-3.

Example 12-3   HELP command

```
==> help             // for basic help
==> 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 in Example 12-4.

Example 12-4   QUERY NAMES

```
==> 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) use the **`QUERY STORAGE`** and **`QUERY XSTOR`** commands shown in Example 12-5.

Example 12-5   QUERY STORAGE and QUEARY XSTOR

```
==> q stor
STORAGE = 3G
==> q xstor
XSTORE= 1024M online= 1024M
XSTORE= 1024M userid= SYSTEM usage= 97% retained= 0M pending= 0M
XSTORE MDC min=0M, max=1024M, usage=96%
XSTORE= 1024M userid=  (none)  max. attach= 1024M
```

Determining processors or CPUs

To see how many processors (CPs, IFLs, CPUs) you have, use the **`QUERY PROCESSORS`** command, shown in Example 12-6 on page 191.

Example 12-6 QUERY PROCESSORS

```
==> q proc
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
PROCESSOR 02 ALTERNATE
PROCESSOR 03 ALTERNATE
```

Determining software level

To determine the level of CP your system, use the **`QUERY CPLEVEL`** command, snow hin Example 12-7.

Example 12-7 QUERY CPLEVEL

```
==> q cplevel
z/VM Version 5 Release 1.0, service level 0401 (64-bit)
Generated at 08/31/04 17:33:32 EST
IPL at 03/10/05 14:42:02 EST
```

Determining system cylinder allocation

The **`QUERY ALLOC MAP`** command in Example 12-8 shows you the system's allocation of spool, paging, and directory space.

Example 12-8 QUERY ALLOC MAP

```
==> q alloc map
                EXTENT     EXTENT                         % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
520RES A770          1         20     20      1      1   5% DRCT ACTIVE
520SPL A773          0       3338 601020  40950  63360   6% SPOOL
520PAG A774          0       3338 601020   8167   9872   1% PAGE
VPA776 A776          0       3338 601020   7840  12448   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 will report on the OSA resources. Finally, the **`QUERY VIRTUAL ALL`** command can be useful. Example 12-9 lists the short form of these commands without any output.

Example 12-9 QUERY DASD and QUERY DASD FREE

```
==> q da
==> q da free
==> q osa
==> q osa free
==> q v all
```

12.2 The z/VM Performance Toolkit

To use the z/VM Performance Toolkit, the product must be ordered. You should only configure the product if you have ordered it.

Much more detail can be found in the following books:

- *z/VM Performance Toolkit*, SC24-6136, on the Web starting at the z/VM 5.2 bookshelf:

  `http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/zvmpdf/zvm52.html`

  Search for **Toolkit** on that page.
- *The Program Directory for Performance Toolkit for VM*, GI11-2854-00, on the Web at

  `http://www.vm.ibm.com/progdir/5vmptk20.pdf`
- The IBM Redbook *Linux on IBM zSeries and S/390®: Performance Toolkit for VM*, SG24-6059, on the Web at:

  `http://www.redbooks.ibm.com/abstracts/sg246059.html`

The sections that follow describe how to set up and use the IBM Performance Toolkit *very briefly:*

- 12.2.1, "Configuring the z/VM Performance Toolkit" on page 192
- 12.2.4, "Using the z/VM Performance Toolkit" on page 195

12.2.1 Configuring the z/VM Performance Toolkit

The Performance Toolkit is installed with z/VM. Configuration is described in the Program Directory. This is a summary of how to turn it on. Again, you should only configure the product if you have ordered it.

To enable it, **logon to `MAINT`** and enter the following command in Example 12-10.

Example 12-10 Configuring the Performance Toolkit

```
==> 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 here.

Alternatively, you can edit the `SYSTEM CONFIG` file and uncomment the last two lines changing `DISABLED` to **`ENABLED`** and reIPL your system.

```
PRODUCT PRODID 5VMPTK10 STATE DISABLED DESCRIPTION '00/00/00.00:00:00.$BASEDDR P
ERFORMANCE TOOLKIT FOR VM'
```

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.

1. Logon to `TCPMAINT`. Edit the `<systemID> TCPIP D` file and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved

   ```
   ==> x <systemID> tcpip d
   ====> /reserve port
   ```

2. Add the following line under the PORT entries:

   ```
   69   UDP TFTPD             ; TFTPD (Trivial FTP) Server
   81   TCP PERFSVM           ; Performance Toolkit
   111  TCP PORTMAP           ; Portmap Server
   ```

3. Save your changes. The `TCPIP` user ID needs to be recycled in order for our changes to take effect. You can **`FORCE`** and **`XAUTOLOG TCPIP`** from a console, or if you are at an emulator session you can reIPL z/VM:

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

When the system comes back, logon to `TCPMAINT` and check if everything was successful by issuing the **`NETSTAT CLIENTS`** command. You want to see that the service `PERFSVM` is a client (listening). This should be shown after a few screens of output:

Example 12-11 Using NETSTAT CLIENTS to check for PERFSVM

```
==> netstat clients
...
Client: PERFSVM                 Authorization: {none}
Notes Handled: none
Last Touched:   0:00:25
Vmcf error count: 0
...
```

The output of the command is lengthy, you want to make sure you have an entry for `PERFSVM`.

12.2.3 Configuring PERFSVM

The `PERFSVM` user ID is the Performance Toolkit service machine.

1. **Logon to `PERFSVM`**. If you successfully enabled the product, you should be put in a Performance Toolkit session and see the following text at the top of the screen:

```
FCX001                Performance Toolkit for VM
FCXBAS500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

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**:

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

   b. Copy the `PROFILE XEDIT` to the A disk:

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

4. Make a backup of the original `PROFILE EXEC` and edit it.

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

5. Copy the default configuration files, which are on `PERFSVM`'s D disk, to your A disk:

```
==> copy * * d = = a
```

6. 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 `*`. Perform the following changes:
   - Uncomment the second and fourth line by changing `*C` to **`FC.`**
   - Change `IDTEST PASSFILE` to `IDTEST` **`CP`** on the fourth line.

– Add the text **`FC MONCOLL LINUXUSR ON`** after the fourth line.

The modified lines should like similar to Example 12-12.

Example 12-12 Editing FCONX $PROFILE

```
==> 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 81 IDTEST CP
FC MONCOLL LINUXUSR ON
*    Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
```

7. Save your changes. The line you added tells the Performance Toolkit to collect Linux performance data.
8. Create a remote data retrieval authorization file – replace `<systemID>` with your system name:

```
==> x fconrmt authoriz
====> a 2
<systemID>  PERFSVM  S&FSERV
<systemID>  MAINT    DATA CMD EXCPMSG
```

9. Create a system identification file – replace `<systemID>` with your system name:

```
==> x fconrmt systems
====> a
<systemID>   PERFSVM  ESA   N   FCXRES00
```

10. Create a Linux system definition file. Add the TCP/IP addresses of your Linux system(s). The following example shows adding two Linux virtual servers, `LINUX01` and `LINUX02`:

```
==> x fconx linuxusr
====> a 2
LINUX01 129.40.178.131:8803
LINUX02 129.40.178.132:8803
```

11. Edit the `PROFILE EXEC` file and uncomment the five `MONITOR SAMPLE` and the two `MONITOR EVENT` statements, as in Example 12-13.

Example 12-13 MONITAOR SAMPLE and MONITOR EVENT statements

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 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 APPLDATA ALL'
'CP MONITOR EVENT  ENABLE STORAGE'
'CP MONITOR EVENT  ENABLE I/O ALL'

'PERFKIT'                    /* Invoke the PERFKIT module   @FC012BD*/

Exit
```

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

You should now be ready to run the Performance Toolkit.

12.2.4 Using the z/VM Performance Toolkit

The Performance Toolkit can be used via 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 81. For example:

   `http://129.40.178.124:81`

2. You should see your system on the Web Session Setup screen. Click it and you will be presented with the Web Server Logon screen.
3. Enter any valid user ID and password. `MAINT` can be used, but need not be.
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 screen as shown in Figure 12-1 on page 196.

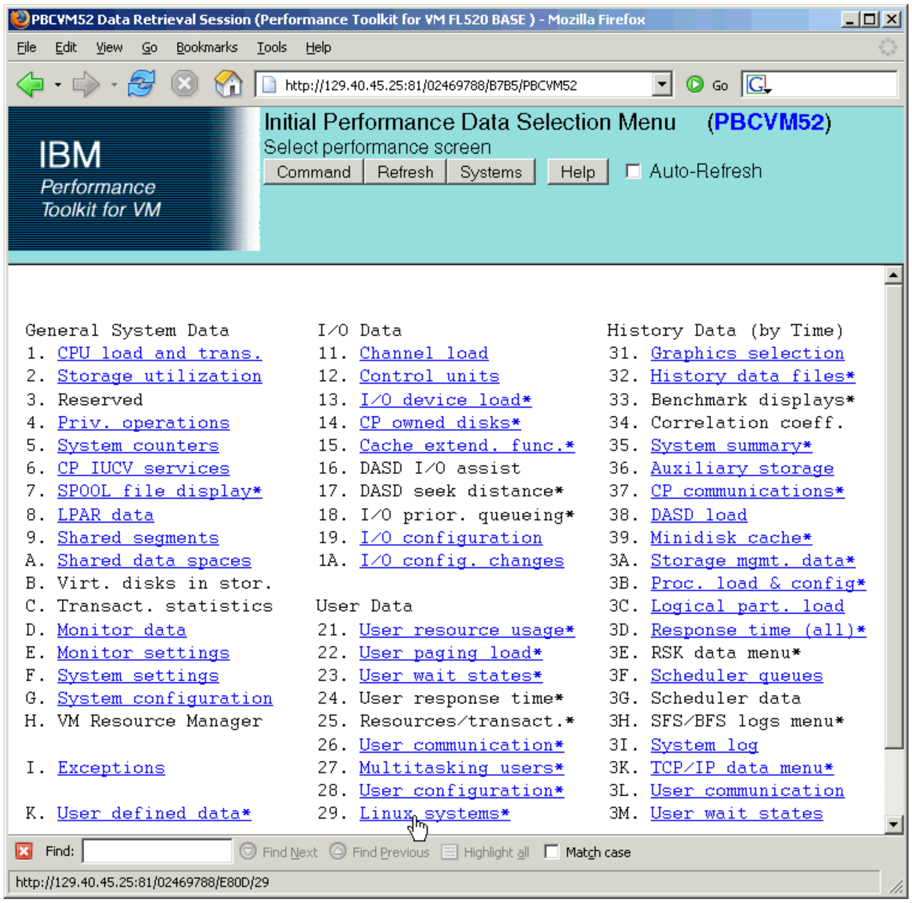


Figure 12-1 Browser interface to the Performance Toolkit

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 panel shown here Example 12-14 and Figure 12-2 on page 197.

Example 12-14 Using PROFILE EXEC

```
==> profile
FCXBAS500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
...
FCX001                  Performance Toolkit for VM
  FCXBAS500I Performance Toolkit for VM FL510 BASE
  HCPMOF6229E Monitor event collection is already active.
  HCPMOG6229E Monitor sample collection is already active.

Command ==> monitor
```

Figure 12-2 Performance Screen Selection

```
FCX124         Performance Screen Selection  (FL510 BASE   )    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. Storage subpools       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. Paging activity
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
```

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`

12.3 Monitoring Linux

To monitor Linux performance data, a data gatherer process must be running. There are different ways of gathering this data. An important distinction is whether the data gathering will be done in the kernel or as a user application. SuSE SLES9 has 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:

```
Linux on zSeries -

    * 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 using 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
```

Following is an example of downloading the tar file for 31-bit distributions directly from the Internet. Open an SSH session on the controller or any other virtual server. Change to the `/opt/` directory and download the appropriate tar file with the **`wget`** command. This example is for a 64-bit distribution:

```
# cd /opt
# wget ftp://ftp.software.ibm.com/eserver/zseries/zos/rmf/rmfpms_s390x_kernel26.tgz
...
13:06:14 (827.64 KB/s) - `rmfpms_s390x_kernel26.tgz' saved [730,332]
```

Untar the file with the **`tar`** command and change to the `rmfpms/` directory, as in Example 12-15.

Example 12-15 tar command

```
# tar xzf rmfpms_s390_kernel26.tgz
# cd rmfpms/
# ls -aF
./   .rmfpms_config            README           bin/  enable_autostart*
../  .rmfpms_config_autostart  autostart_rmfpms doc/
```

You should see the configuration file.`rmfpms_config`. Make a copy of the configuration file, edit it and change `$HOME` to `/opt` and save your changes, as in Example 12-16.

Example 12-16 Copying and editing rmfpms_config

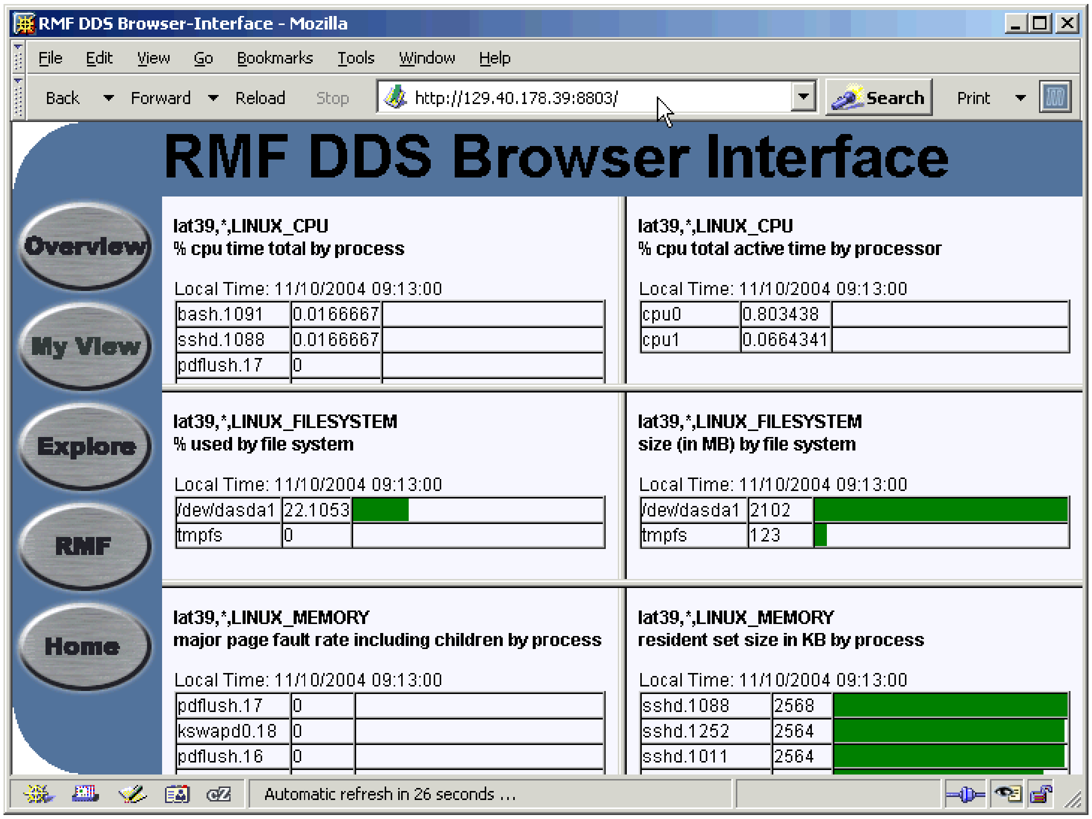
```
# cp .rmfpms_config .rmfpms_config.orig
# vi .rmfpms_config      // modify two lines with $HOME, commend out APACHE variables:
# rmfpms_config - included in rmfpms bash shell script
#
# 11/14/2000, 11/11/2004 Oliver Benke
# (c) IBM Deutschland Entwicklung GmbH, IBM Corp.
#
# configuration parameters
#
#
export RMFPMS_HOME=/opt/rmfpms
#
...
```

You can now start **rmfpms** in the `bin/` directory with the following command in Example 12-17.

Example 12-17 Using rmfpms in the bin/ directory

```
# bin/rmfpms start
Creating /opt/IBM/rmfpms/.rmfpms ...
Starting performance gatherer backends ...
 DDSRV: RMF-DDS-Server/Linux-Beta (Aug  9 2004) started.
 DDSRV: Functionality Level=2.008
 DDSRV: Reading exceptions from gpmexsys.ini and gpmexusr.ini.
DDSRV: Server will now run as a daemon process.
done!
```

When it is running, you can view the performance data from a browser pointing to the Linux image and port 8803 as shown in Figure 12-3. You can also register Linux images with the Performance Toolkit. See section 12.4, “Registering Linux images with the Performance Toolkit” on page 200.



lat39,*,LINUX_CPU
% cpu time total by process

Local Time: 11/10/2004 09:13:00

bash.1091	0.0166667
sshd.1088	0.0166667
pdflush.17	0

lat39,*,LINUX_CPU
% cpu total active time by processor

Local Time: 11/10/2004 09:13:00

cpu0	0.803438
cpu1	0.0664341

lat39,*,LINUX_FILESYSTEM
% used by file system

Local Time: 11/10/2004 09:13:00

/dev/dasda1	22.1053
tmpfs	0

lat39,*,LINUX_FILESYSTEM
size (in MB) by file system

Local Time: 11/10/2004 09:13:00

/dev/dasda1	2102
tmpfs	123

lat39,*,LINUX_MEMORY
major page fault rate including children by process

Local Time: 11/10/2004 09:13:00

pdflush.17	0
kswapd0.18	0
pdflush.16	0

lat39,*,LINUX_MEMORY
resident set size in KB by process

Local Time: 11/10/2004 09:13:00

sshd.1088	2568
sshd.1252	2564
sshd.1011	2564

Automatic refresh in 26 seconds ...

Figure 12-3 Browser view of rmfpms-gathered Linux data

12.3.2 Monitoring Linux performance data from the kernel

To monitor Linux performance data directly from the kernel, the following 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 as the `OPTION APPLMON` was set for SLES9 in section 7.1, “Creating the user ID SLES9” on page 86, and for the other Linux user IDs initially in section 9.1, “Defining a new user ID for an virtual server” on page 132.

With regards to the second requirement, SuSE SLES9 now has this function built in. Details of this function are described in Chapter 15, “Linux monitor stream support for z/VM” in the manual *Device Drivers, Features, and Commands* for the October 2005 stream, on the Web:

`http://www-128.ibm.com/developerworks/linux/linux390/october2005_documentation.html`

A quick description of how to use this built-in monitoring function follows.

There are three modules that are built into the kernel but are not loaded by default. They are named `appldata_mem`, `appldata_os` and `appldata_net_sum`. You can verify that they are not loaded with the **`lsmod`** and **`grep`** commands:

```
# lsmod | grep appldata
```

There is no output, so no modules with the string `appldata` are loaded. Load those modules now with the **`modprobe`** command and verify they have been loaded:

```
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

Now if you repeat the **`lsmod`** command, you should see the following:

```
# lsmod | grep appldata
appldata_net_sum       20064  0
appldata_os            21512  0
appldata_mem           20112  0
```

The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory there are five files as follow:

`timer`	Controls whether any data gathering is in effect.
`interval`	Sets the interval, in milliseconds, that samples will be taken.
`mem`	Controls the memory data gathering module.
`os`	Controls the CPU data gathering module.
`net_sum`	Controls the net data gathering module.

To turn on the built in kernel monitoring, use the **`echo`** command to send a non-zero value into four of the five monitoring variables in the `/proc/` virtual file system:

```
# echo 1 > /proc/sys/appldata/timer
# echo 1 > /proc/sys/appldata/mem
# echo 1 > /proc/sys/appldata/os
# echo 1 > /proc/sys/appldata/net_sum
```

Built-in kernel monitoring should now be turned on. You might only want to leave the monitoring on for specific periods of time. As Linux monitoring data is captured, the Performance Toolkit’s minidisk space can fill up relatively quickly.

12.4 Registering Linux images with the Performance Toolkit

To register Linux images that have performance data gathering enabled, **logon to `PERFSVM`** and create a file `FCONX LINUXUSR A` and add Linux user ID/IP address:port pairs. Example 12-18 shows adding the `SLES9` and `LINUX01` user IDs to this file.

Example 12-18 Adding user IDs to FCONX LINUXUSR

```
==> x fconx linuxusr a
====> a 3
*Linux-ID    IP address for DDS Interface:Port
SLES9X       <129.40.178.127>
```

```
   LINUX01       <129.40.178.131>
```

Restart the performance toolkit:

```
==> profile
```

After the system has had some time to collect data, you should be able to use the Performance Toolkit to monitor Linux systems that have both monitoring data being captured and an entry in the `FCONX LINUXUSR` file. To view that data, drill down into menu 29, `Linux systems`.

13

Backup and restore

This chapter addresses backing up and restoring systems. Given the opening quote, Albert Einstein would probably have not have made a good z/VM and Linux system administrator :))

Backup and restore can be divided into two fundamental issues. Each of these two issues must be able to answer a basic question:

Incremental back up	How do I quickly get back a file that was accidentally deleted or corrupted?
Disaster recovery	How do I restore my infrastructure if my entire data center is wiped out?

The key to incremental back up is self-service. Ideally, you as the system administrator should not have to restore individual files that have been deleted or corrupted, rather, you should be easily be able to tell end users how to do it themselves.

The key to disaster recovery is to have a plan and to practice it. There is no *silver bullet*. It takes planning and work. You should practice restoring your system at least twice a year, preferable more often. One rule of disaster recovery to consider is “For every backup you make, you should do one restore”.

Both your z/VM system and your Linux servers must be considered. Therefore, the following four permutations exist and are at least briefly addressed in this chapter:

- “Incremental backup of z/VM” on page 204
- “Incremental backup of Linux servers” on page 206
- “Disaster recovery of z/VM and virtual Linux servers” on page 208

13.1 Incremental backup of z/VM

If you completed section 4.9, “Backing up your z/VM system to tape” on page 56 and perhaps even 4.11, “Restoring your z/VM system from tape” on page 61, then you have a copy of your system after it was first customized with networking. Since that time, you have customized at least the `USER DIRECT` file and possibly some of the other important configuration files on z/VM:

- `SYSTEM CONFIG` on the `MAINT CF1` minidisk
- `USER DIRECT` on the `MAINT 2CC` minidisk
- `<system_ID> TCPIP` and `SYSTEM DTCPARMS` on the `TCPMAINT 198` minidisk
- `TCPIP DATA` on the `TCPMAINT 592 minidisk`
- `PROFILE EXEC` on the `AUTOLOG1 191 minidisk`

You might find you can get access to your z/VM system, but for some reason might have lost or corrupted one or more of these files. Having nightly copies of them can get you out of a tight spot (remember that all the z/VM passwords are in the `USER DIRECT` file, so if back this file up be sure the directory where it is stored is secure).

In addition, it would be helpful to backup all files on the `LNXMAINT 192` minidisk.

A simple script named **`backup_vm`** has been included on the controller in the /etc/cron.daily/ directory to help you back up these files. This script should not be your only backup procedure, but it can help if you are in a pinch.

13.1.1 The backup_vm script

> **Important:** This script backs up important z/VM data to a Linux system running under the same z/VM. If your z/VM system cannot be brought back, then this Linux system cannot be started. Therefore, consider running this script on a different LPAR or different physical server.

One way to back these up nightly are with the script in Example 13-1 and corresponding input files. The script uses the **`ftp`** client to login to z/VM, four times for each of the MAINT, TCPMAINT, AUTOLOG1 and LNXMAINT user IDs. Note that the TCP/IP address of z/VM and the passwords to `MAINT`, `TCPMAINT` and `AUTOLOG1` are hard-coded into the script:

Example 13-1 Using backup_vm

```
# cd /etc/cron.daily/
# tail -7 backup_vm
cd /backup/vm
getVMinfo > zVMinfo.txt
ftp ftp://maint:<lnx4vm>@<129.40.178.124> < /etc/cron.daily/MAINT.FTPcommands
ftp ftp://tcpmaint:<lnx4vm>@<129.40.178.124> < /etc/cron.daily/TCPMAINT.FTPcommands
ftp ftp://autolog1:<lnx4vm>@<129.40.178.124> < /etc/cron.daily/AUTOLOG1.FTPcommands
cd LNXMAINT
ftp ftp://lnxmaint:<lnx4vm>@<129.40.178.124> < /etc/cron.daily/LNXMAINT.FTPcommands
```

Edit the file and modify the four passwords and four TCP/IP addresses.

```
# vi backup_vm       // modify the four lines that are shown in bold above
...
```

There are also four FTP command files in the same directory.

The FTP command files

The `MAINT.FTPcommands` file in Example 13-2 backs up the `USER DISKMAP`, `USER DIRECT` and `SYSTEM CONFIG` files.

Example 13-2 Using MAINT.FTP

```
# cat MAINT.FTPcommands
ascii
get USER.DISKMAP
cd maint.2cc
get USER.DIRECT
cd maint.cf1
get SYSTEM.CONFIG
quit
```

The `TCPMAINT.FTPcommands` file in Example 13-3 backs up the `<system_ID> TCPIP`, `SYSTEM DTCPARMS` and `TCPIP DATA` files. The z/VM system name hard coded into this file must be modified:

Example 13-3 Using TCPMAINT.FTP

```
# cat TCPMAINT.FTPcommands
ascii
cd tcpmaint.198
get <LNXVM52>.TCPIP
get SYSTEM.DTCPARMS
cd tcpmaint.592
get TCPIP.DATA
quit
# vi TCPMAINT.FTPcommands    // modify the line with <LNXVM52>.TCPIP
...
```

The `AUTOLOG1.FTPcommands` file backs up the `AUTOLOG1 PROFILE EXEC` as the file `PROFILE.EXECAUT1`:

```
# cat AUTOLOG1.FTPcommands
ascii
get PROFILE.EXEC PROFILE.EXECAUT1
quit
```

The `LNXMAINT.FTPcommands` file backs up files on the `LNXMAINT 192` disk as they would exist at the end of the steps in this book. If additional files are added, this file might have to be modified depending upon the file names. Example 13-4 is an example set of FTP commands that can be used to backup files off the LNXMAINT 192 disk:

Example 13-4 FTP commands for backing up LNXAINT 192 disk files

```
# cat LNXMAINT.FTPcommands
ascii
cd lnxmaint.192
prompt
mget C*
mget D*
mget LINUX*
mget P*
mget SLES*
mget VSW*
quit
```

The getVMinfo script

There are some CP commands which also capture important information about the systems. A small script, **`getVMinfo`** is supplied to issue the CP commands (Example 13-5). This file should have been copied to the directory `/usr/local/sbin/`.

Example 13-5 Using getVMinfo

```
# cat /usr/local/sbin/getVMinfo
#!/bin/bash
#
# getVMinfo - run 4 CP commands to report some useful system info
#
# -----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
...#
echo "output of CP QUERY PROC:"
echo "------------------------"
hcp QUERY PROC
echo ""
echo "output of CP QUERY FILES:"
echo "-------------------------"
hcp QUERY FILES
echo ""
echo "output of CP QUERY NSS ALL MAP:"
echo "-------------------------------"
hcp QUERY NSS ALL MAP
echo ""
echo "output of CP QUERY CPLEVEL:"
echo "---------------------------"
hcp QUERY CPLEVEL
```

> **Attention:** Another more sophisticated script, **`VMquery`**, was written that also allows CMS commands to be used. That script is not described here, but you can investigate it. It also should be in the `/usr/local/sbin/` directory.

Because the **`backup_vm`** script is in the `/etc/cron.daily/` directory on the controller, it should run automatically once a day. Try running it now:

```
# cd /backup/vm
# /etc/cron.daily/backup_vm
```

You should see a large number of files being copied via ftp. When you are done investigate the files. All text files are converted to ASCII so you should be able to read them:

```
# ls -F
./   LNXMAINT/      PROFILE.EXECAUT1  SYSTEM.DTCPARMS  USER.DIRECT   zVMinfo.txt
../  LNXVM52.TCPIP  SYSTEM.CONFIG     TCPIP.DATA       USER.DISKMAP
```

13.2 Incremental backup of Linux servers

Your IT department might have an existing backup/restore solution. For example:

- Computer Associates Brightstor Enterprise Backup
- Innovation Data Processing FDR/UPSTREAM
- IBM Tivioli Storage Manager (TSM)
- SecureAgent Software SecureBackup
- Veritas Backup Exec

For details of these and other solutions, see the *Software Developer Products for Linux on zSeries and S/390* page on the Web at:

`http://www-1.ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html`

Descriptions of these products are outside the scope of this book, and they are a much more complete solution than what is described in the sections that follow.

Some simple **cron** scripts and configuration of **rsync** follow. These will result in the `/etc/` directories and other information of the virtual servers (clones) being backed up to the controller nightly.

13.2.1 Configuring rsync on the controller

Because the master image was set up to allow the controller to copy its `/etc/` directory, it can be copied with a single **rsync** command. For example, the following commands will copy the `/etc/` directory on `LINUX01` at IP address 129.40.178.131:

```
# cd /backup/LINUX01-on-<129.40.178.131>
# rsync -r --timeout=30 <129.40.178.131>:/etc .
```

A script named **backup_linux** (Example 13-6) is provided to back up the `/etc/` directory of each virtual server nightly. It is invoked nightly by being in the directory `/etc/cron.daily/` and iterates over the directories created by the clone script under `/backup/`.

Example 13-6 Using backup_linux

```
# cd /etc/cron.daily
# tail -9 backup_linux
# modify next line if the rsync backup directory is OTHER than /backup/:
backup_dir="/backup/linux"
cd $backup_dir
for i in LINUX*-on-*  # iterate through directories starting with LINUX*-on-*
do
  IP_addr=${i#LINUX*-on-}  # this chops the head off and grabs the IP address
  cd $backup_dir/$i        # change directory
  rsync -r --timeout=30 $IP_addr:/etc .    # use rsync to back up the /etc dir
done
```

This script should backup the `/etc/` directory of each Linux system cloned to the appropriate directory under `/backup/linux/`.

13.2.2 Configuring sitar to run nightly on the virtual servers

SITAR is a tool that creates documentation describing your system. It is an acronym for System InformaTion At Runtime. In addition to backing up the `/etc/` configuration files, a document describing your system might be helpful. If sitar is run once a day and the output is put in a file in `/etc/`, then you will have additional information about your system backed up.

This command can easily be run nightly via **cron**:

```
# cd /etc/cron.daily
# vi run-sitar    // add two lines
#!/bin/bash
sitar --format=html --outfile=/etc/sitar.html
```

13.3 Disaster recovery of z/VM and virtual Linux servers

In addition to incremental backups, you should do regular system backups of your z/VM system. Sections 4.9, “Backing up your z/VM system to tape” on page 56 and 4.11, “Restoring your z/VM system from tape” on page 61 just touched on this.

You can choose to do a full volume backups of all DASD in the LPAR. Then in the event of a disaster, your entire system could be restored. This is the most thorough and best way to perform backups, and is one of the areas where the zSeries platform excels.

You can also choose to backup a subset of the volumes. Alternatively, perhaps you choose to backup all volumes once a quarter, but back up a subset once a month. Two critical user IDs are `LNXMAINT` and `SLES9`. If you have followed all steps in this book, then each of the virtual servers `/etc/` file system is backed up to the `/backup/` file system. This file system is on `/dev/dasde1` which is the `SLES9 104` minidisk. The master image (on `SLES9 100` and `102`) and the controller (on `SLES9 103`) are also critical. Given that these minidisks are vital to your system, you can choose to back them up more frequently or in a different fashion. You can look at the `USER DISKMAP` file, created on the **`MAINT 191`** disk by the **`DISKMAP`** command, to determine which volumes to back up. In the examples in this book, the three volumes `VMA77C`, `VMA77A` and `VMA77B` contain these minidisks in Example 13-7.

Example 13-7 Minidisk volumes created for this book

```
VOLUME   USERID     CUU   DEVTYPE      START        END       SIZE
VMA77C   $ALLOC$    A04     3390       00000      00000      00001
         LNXMAINT   191     3390       00001      00020      00020
         LNXMAINT   192     3390       00021      00320      00300
         DTCVSW1    191     3390       00321      00325      00005
         DTCVSW2    191     3390       00326      00330      00005
         SLES9      104     3390       00331      03333      03003

VOLUME   USERID     CUU   DEVTYPE      START        END       SIZE
VMA77A   $ALLOC$    A05     3390       00000      00000      00001
         SLES9      100     3390       00001      03038      03038
         SLES9      102     3390       03039      03338      00300

VOLUME   USERID     CUU   DEVTYPE      START        END       SIZE
VMA77B   $ALLOC$    A06     3390       00000      00000      00001
         SLES9      103     3390       00001      03038      03038
```

You will probably have many more critical volumes than these three, but consider these especially in your hierarchy of backed-up data.

A

References and source code

This book refers to additional material that can be downloaded from the Internet as described in this appendix.

A.1 Obtaining and using the Web material

The files associated with this book are available in soft copy on the Internet from:

ftp://www.redbooks.ibm.com/redbooks/SG246695/

Download the `virt-cookbook.tgz` file to the NFS server and use it as is described in section 6.1, “Downloading files associated with this book” on page 76. The hierarchy within both of the tar files named `virt-cookbook/`. Under that directory are the following file and directories:

`README.txt`	The main README file
`linux-controller/`	Files used on Linux *controller*
`linux-master/`	Files used on Linux *master image* which will become cloned servers
`nfs-server/`	Files used on the temporary NFS server
`vm/`	Files used on z/VM

A.2 Related publications

These publications are also relevant as further information sources:

- *Linux for zSeries and S/390 Device Drivers, Features, and Commands*, LNUX-1403
- SUSE LINUX Enterprise Server: INSTALLATION AND ADMINISTRATION
- SUSE LINUX Enterprise Server: ARCHITECTURE-SPECIFIC INFORMATION
- SUSE LINUX Enterprise Server: START-UP GUIDE
- *z/VM Guide for Automated Installation and Service: Version 5 Release 1.0*, GC24-6099
- *z/VM System Messages and Codes — CP: Version 5 Release 1.0*, GC24-6119
- *z/VM TCP/IP Messages and Codes: Version 5 Release 1.0*, GC24-6124
- *The Program Directory for Performance Toolkit for VM*, GI11-4800
- *z/VM CP Commands and Utilities Reference: Version 5 Release 1.0*, SC24-6081
- *z/VM CP Planning and Administration: Version 5 Release 1.0*, SC24-6083
- *z/VM Getting Started with Linux on zSeries: Version 5 Release 1.0*, SC24-6096
- *z/VM TCP/IP Planning and Customization: Version 5 Release 1.0*, SC24-6125
- *z/VM Performance Toolkit*, SC24-6136
- *Communication Controller for Linux on zSeries V1.0 Implementation Guide*, SC31-6872
- *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
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864

A.3 Online resources

These Web sites and URLs 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 for zSeries 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/

A.4 Important z/VM 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 13-1provides a summary and the location of important z/VM configuration files.

Table 13-1 Important 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.

A.5 z/VM source code listings

This section lists the contents of the REXX EXECs and XEDIT macros described in the book.

A.5.1 The CPFORMAT EXEC

Example A-1 is the code for the EXEC that formats multiple disks with **CPFMTXA** (described in section 4.5.1, "Formatting the paging volumes" on page 42):

Example: A-1 CPFMTXA code

```
/*+------------------------------------------------------------------+*/
/*| 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   |*/
/*+------------------------------------------------------------------+*/

/*-------------------------------------------------------------------
```

```
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 = 'V' /* change this for an LPAR ID other than 'V' */
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
  retVal = doFormat(labelPrefix numDasd type)
  call doReport
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 and 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 <> '')
    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.
/*| 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)
    call formatOne(dasdList.i type label)
  end /* do i = */
  'CP TERM MORE 50 10'
return 0 /* 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 0190 0191 0193 0194 019D 019E 0201 02A2"
  resvd2 = "02A4 02A6 02C2 02C4 02CC 02D2 0319 03A2 03A4 03A6 03B2 03C2"
  resvd3 = "03C4 03D2 0400 0401 0402 0405 0490 0493 049B 049E 04A2 04A4"
  resvd4 = "04A6 04B2 04C2 04C4 04D2 0500 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
/*| Report on the newly labelled DASD                                |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  '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'
return /* 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 */
```

A.5.2 The CHPW52 XEDIT macro

Example A-2 is the code for the XEDIT macro that changes all passwords in the z/VM 5.2 `USER DIRECT` file:

Example: A-2 XEDIT macro code

```
/* CHPW52 XEDIT - change all passwords in z/VM 5.2 USER DIRECT file */

/*-------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
...
-------------------------------------------------------------------*/

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/CMS1  CMS1/CMS1' newPass'/*'
'command c/LGLOPR  LGLOPR/LGLOPR' newPass'/*'
'command c/MAINT MAINT/MAINT' newPass'/*'
'command c/CMSBATCH CMSBATCH/CMSBATCH' newPass'/*'
'command c/AVSVM AVSVM/AVSVM' newPass'/*'
'command c/TSAFVM TSAFVM/TSAFVM' newPass'/*'
'command c/VMSERVS VMSERVS/VMSERVS' newPass'/*'
'command c/VMSERVU VMSERVU/VMSERVU' newPass'/*'
'command c/VMSERVR VMSERVR/VMSERVR' newPass'/*'
'command c/GCS GCS/GCS' newPass'/*'
'command c/GCSXA GCSXA/GCSXA' newPass'/*'
'command c/SYSMAINT SYSMAINT/SYSMAINT' newPass'/*'
'command c/OPERATOR OPERATOR/OPERATOR' newPass'/*'
'command c/OP1 OP1/OP1' newPass'/*'
'command c/EREP EREP/EREP' newPass'/*'
'command c/OPERATNS OPERATNS/OPERATNS' newPass'/*'
'command c/AUTOLOG1 AUTOLOG1/AUTOLOG1' newPass'/*'
'command c/DISKACNT DISKACNT/DISKACNT' newPass'/*'
'command c/OPERSYMP OPERSYMP/OPERSYMP' newPass'/*'
'command c/VMUTIL VMUTIL/VMUTIL' newPass'/*'
'command c/SYSDUMP1 SYSDUMP1/SYSDUMP1' newPass'/*'
'command c/5684042J 5684042J/5684042J' newPass'/*'
'command c/4OSASF40 4OSASF40/4OSASF40' newPass'/*'
'command c/OSASF OSASF/OSASF' newPass'/*'
'command c/OSAMAINT OSAMAINT/OSAMAINT' newPass'/*'
'command c/OSADMIN1 OSADMIN1/OSADMIN1' newPass'/*'
'command c/OSADMIN2 OSADMIN2/OSADMIN2' newPass'/*'
'command c/OSADMIN3 OSADMIN3/OSADMIN3' newPass'/*'
'command c/P684096K P684096K/P684096K' newPass'/*'
'command c/RSCS RSCS/RSCS' newPass'/*'
'command c/XCHANGE XCHANGE/XCHANGE' newPass'/*'
'command c/RSCSDNS RSCSDNS/RSCSDNS' newPass'/*'
'command c/TCPMAINT TCPMAINT/TCPMAINT' newPass'/*'
'command c/5VMTCP20 5VMTCP20/5VMTCP20' newPass'/*'
'command c/TCPIP TCPIP/TCPIP' newPass'/*'
'command c/IMAP IMAP/IMAP' newPass'/*'
'command c/FTPSERVE FTPSERVE/FTPSERVE' newPass'/*'
'command c/SMTP SMTP/SMTP' newPass'/*'
'command c/NAMESRV NAMESRV/NAMESRV' newPass'/*'
```

```
'command c/REXECD REXECD/REXECD' newPass'/*'
'command c/RXAGENT1 RXAGENT1/RXAGENT1' newPass'/*'
'command c/X25IPI X25IPI/X25IPI' newPass'/*'
'command c/PORTMAP PORTMAP/PORTMAP' newPass'/*'
'command c/NDBPMGR NDBPMGR/NDBPMGR' newPass'/*'
'command c/NDBSRV01 NDBSRV01/NDBSRV01' newPass'/*'
'command c/SNMPQE SNMPQE/SNMPQE' newPass'/*'
'command c/SNMPD SNMPD/SNMPD' newPass'/*'
'command c/ROUTED ROUTED/ROUTED' newPass'/*'
'command c/LPSERVE LPSERVE/LPSERVE' newPass'/*'
'command c/SNALNKA SNALNKA/SNALNKA' newPass'/*'
'command c/VMNFS VMNFS/VMNFS' newPass'/*'
'command c/VMKERB VMKERB/VMKERB' newPass'/*'
'command c/ADMSERV ADMSERV/ADMSERV' newPass'/*'
'command c/UFTD    UFTD/UFTD' newPass'/*'
'command c/BOOTPD BOOTPD/BOOTPD' newPass'/*'
'command c/TFTPD TFTPD/TFTPD' newPass'/*'
'command c/DHCPD DHCPD/DHCPD' newPass'/*'
'command c/MPROUTE MPROUTE/MPROUTE' newPass'/*'
'command c/SSLSERV SSLSERV/SSLSERV' newPass'/*'
'command c/MONWRITE MONWRITE/MONWRITE' newPass'/*'
'command c/5VMDIR10 5VMDIR10/5VMDIR10' newPass'/*'
'command c/BLDNUC  BLDNUC/BLDNUC' newPass'/*'
'command c/BLDCMS  BLDCMS/BLDCMS' newPass'/*'
'command c/AUDITOR AUDITOR/AUDITOR' newPass'/*'
'command c/SYSMON  SYSMON/SYSMON' newPass'/*'
'command c/VMRMSVM VMRMSVM/VMRMSVM' newPass'/*'
'command c/VMRMADMN VMRMADMN/VMRMADMN' newPass'/*'
'command c/5767002P 5767002P/5767002P' newPass'/*'
'command c/RACFVM RACFVM/RACFVM' newPass'/*'
'command c/RACFSMF RACFSMF/RACFSMF' newPass'/*'
'command c/RACMAINT RACMAINT/RACMAINT' newPass'/*'
'command c/AUTOLOG2 AUTOLOG2/AUTOLOG2' newPass'/*'
'command c/IBMUSER IBMUSER/IBMUSER' newPass'/*'
'command c/SYSADMIN SYSADMIN/SYSADMIN' newPass'/*'
'command c/BLDRACF  BLDRACF/BLDRACF' newPass'/*'
'command c/5VMPTK20 5VMPTK20/5VMPTK20' newPass'/*'
'command c/PERFSVM  PERFSVM/PERFSVM' newPass'/*'
'command c/5VMHCD20 5VMHCD20/4VMHCD40' newPass'/*'
'command c/CBDIODSP CBDIODSP/CBDIODSP' newPass'/*'
'command c/VSMSERVE VSMSERVE/VSMSERVE' newPass'/*'
'command c/IMAPAUTH IMAPAUTH/IMAPAUTH' newPass'/*'
'command c/DTCVSW1  DTCVSW1/DTCVSW1 ' newPass'/*'
'command c/DTCVSW2  DTCVSW2/DTCVSW2 ' newPass'/*'
'command c/MIGMAINT MIGMAINT/MIGMAINT' newPass'/*'
'command c/LNXMAINT LNXMAINT/LNXMAINT' newPass'/*'
'command c/BLDSEG BLDSEG/BLDSEG' 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/RBOOTPD  WBOOTPD  MBOOTPD/'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/RNDBSRV0 WNDBSRV0 MNDBSRV0/'newPass newPass newPass'/*'
'command c/RPORTMAP WPORTMAP MPORTMAP/'newPass newPass newPass'/*'
'command c/RREXECD  WREXECD  MREXECD/'newPass newPass newPass'/*'
'command c/RROUTED  WROUTED  MROUTED/'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/MR READ/'MR newPass'/*'
```

A.5.3 The LABEL520 EXEC

Example A-3 is the code for the EXEC that changes the system labels of a z/VM 5.2 system:

Example: A-3 EXEC code for changing z/VM 5.2 system labels

```
/*+------------------------------------------------------------------+*/
/*| EXEC: LABEL520  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   |*/
/*+------------------------------------------------------------------+*/

/*------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
...
-------------------------------------------------------------------*/
```

```
parse upper arg res spl pag w01 w02 .
if (w02 = '') then call help

/* Construct the two character label prefix */
firstChar = 'V' /* change this for an LPAR ID other than 'V' */
labelPrefix = firstChar'V'

/* 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 520PAG 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
/*+------------------------------------------------------------------+*/
  parse source . . fn .
  say ""
  say "Synopsis:"
  say ""
  say "Relabel the five system volumes (520RES, 520W01, ...) to VV<xxxx>"
  say "   where <xxxx> is the 4 digit address"
  say ""
  say "Syntax is:"
  say ""
  say "   >>---LABEL520--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 */
```

A.5.4 The LABEL520 XEDIT macro

Example A-4 is the code for the XEDIT macro that changes all passwords in the z/VM 5.2 `USER DIRECT` file:

Example: A-4 XEDIT macro for chaning passwords

```
/* LABEL520 - change '520xxx' labels in SYSTEM CONFIG or USER DIRECT */
/*-----------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
...
-----------------------------------------------------------------*/

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/520RES/VV'res'/*'
'command c/520W01/VV'w01'/*'
'command c/520W02/VV'w02'/*'
'command c/520SPL/VV'spl'/*'
'command c/520PAG/VV'pag'/*'
```

A.6 Linux source code listings

This section lists the contents of the two shell scripts described in the book.

A.6.1 The mksles9root.sh script

This section lists the **`mksles9root.sh`** script in Example A-5. See section 6.2.5, “Creating the SLES9 install tree” on page 80 for a description:

Example: A-5 mksles9root.sh script

```
#!/bin/bash
#
# mksles9root.sh - a script to build a SLES9 install tree
# ---------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
...
# ---------------------------------------------------------------------------
#

#+--------------------------------------------------------------------------+
function usage
# Give help
#+--------------------------------------------------------------------------+
 {
  scriptName=`basename $0`
  echo ""
  echo "$scriptName: Create a SLES9 install tree "
  echo ""
  echo "Usage: $scriptName [option] <arch>"
  echo "where [option] can be:"
  echo "    -m, --mounts "
  echo "        Mount ISO images loopback, don't copy"
  echo "where <arch> must be one of:"
  echo "    s390"
  echo "        For 31-bit SLES9 install tree"
  echo "    s390x"
  echo "        For 64-bit SLES9"
  echo ""
  echo "The 6 ISO images for vanilla SLES9 must exist in the current directory"
  echo "If 3 SP3 ISO images exist, they will be built into the tree sles9[x]sp3root/"
  echo "else if 3 SP2 ISO images exist, they will be built into the tree
sles9[x]sp2root/"
  echo "else if 2 SP1 ISO images exist, they will be built into the tree
sles9[x]sp1root/"
  echo "else the vanilla SLES9 tree will be sles9[x]root/"
  exit
 }

#+--------------------------------------------------------------------------+
function check_for_discs
# Check if the 6 SLES9 ISO images exist - if any are missing then exit
# Check for the 2 SLES9 SP1 and the 3 SP2 ISO images
# return:
#     0: Only the 6 SLES9 ISO images are found
#     1: The 2 SP1 ISO images are found
#     2: The 3 SP2 ISO images are found
#     3: The 3 SP3 ISO images are found
#+--------------------------------------------------------------------------+
 {
  for i in $cd1 $cd2 $cd3 $cd4 $cd5 $cd6; do # check for the 6 SLES9 ISO images
    if [ ! -f $i ]; then
      echo "The file $i is not found"
      echo "Error: cannot proceed without the following 6 SLES9 ISO images:"
      echo "$cd1"
      echo "$cd2"
      echo "$cd3"
      echo "$cd4"
      echo "$cd5"
      echo "$cd6"
      echo ""
      usage
    fi
  done
  retVal="0"
  if [[ -f $sp3cd1 && -f $sp3cd2 && -f $sp3cd3 ]]; then
    echo "SP3 ISO images found ..."
    retVal=3
  elif [[ -f $sp2cd1 && -f $sp2cd2 && -f $sp2cd3 ]]; then
    echo "SP2 ISO images found ..."
    retVal=2
  elif [[ -f $sp1cd1 && -f $sp1cd2 ]]; then # SP1 ISOs found
    echo "SP1 ISO images found ..."
    retVal=1
  fi
  return $retVal
 }

#+--------------------------------------------------------------------------+
function mount_copy
# Mount an ISO image over the directory tmpCD/ and copy its contents
#   arg 1: the ISO image to mount
#   arg 2: the temporary directory over which to mount
#   arg 3: the directory to copy to
#+--------------------------------------------------------------------------+
 {
  ISO_image=$1
  temp_dir=$2
  target_dir=$3
  echo "  Mounting and copying $ISO_image ..."
  mount -o loop,ro $ISO_image $temp_dir
  if [ $? != 0 ]; then # unable to mount
    echo "Error: unable to mount $ISO_image over $temp_dir - exiting"
    rmdir $temp_dir
    exit
  fi
  cd $temp_dir
  rsync -HlogptrS * ../$target_dir
  cd ..
  umount $temp_dir
 }
```

```
#+--------------------------------------------------------------------------+
function copy_CDs
# Mount and copy appropriate CDs
#   arg 1: service pack level (0 = vanilla)
#+--------------------------------------------------------------------------+
 {
  echo "Copying SLES9 ISO images ..."
  temp_dir=tmpCD
  if [ ! -d $temp_dir ]; then mkdir $temp_dir; fi
  mount_copy $cd1 $temp_dir $rdir/$sles/CD1/
  mount_copy $cd2 $temp_dir $rdir/$core/CD1/
  mount_copy $cd3 $temp_dir $rdir/$core/CD2/
  mount_copy $cd4 $temp_dir $rdir/$core/CD3/
  mount_copy $cd5 $temp_dir $rdir/$core/CD4/
  mount_copy $cd6 $temp_dir $rdir/$core/CD5/
  if [ $1 = 1 ]; then # then SP1 ISO images exist
    echo "Copying SLES9 SP1 ISO images ..."
    mount_copy $sp1cd1 $temp_dir $rdir/$sp1/CD1
    mount_copy $sp1cd2 $temp_dir $rdir/$sp1/CD2
  elif [ $1 = 2 ]; then # then SP2 ISO images exist
    echo "Copying SLES9 SP2 ISO images ..."
    mount_copy $sp2cd1 $temp_dir $rdir/$sp2/CD1
    mount_copy $sp2cd2 $temp_dir $rdir/$sp2/CD2
    mount_copy $sp2cd3 $temp_dir $rdir/$sp2/CD3
  elif [ $1 = 3 ]; then # then SP3 ISO images exist
    echo "Copying SLES9 SP3 ISO images ..."
    mount_copy $sp3cd1 $temp_dir $rdir/$sp3/CD1
    mount_copy $sp3cd2 $temp_dir $rdir/$sp3/CD2
    mount_copy $sp3cd3 $temp_dir $rdir/$sp3/CD3
  fi
  echo "Removing temporary mount point ..."
  rmdir $temp_dir
 }

#+--------------------------------------------------------------------------+
function mk_directory_structure
# Make the directory structure
#   arg 1: service pack level (0 = vanilla)
# The directory structure is as follows:
# /sles9root/
#     boot -> sles9/CD1/boot
#     content -> sles9/CD1/content
#     control.xml -> sles9/CD1/control.xml
#     media.1 -> sles9/CD1/media.1
#     core9/
#         CD1/
#         CD2/
#         CD3/
#         CD4/
#         CD5/
#     sles9/
#         CD1/
#     sp1-9/        // if the 2 SP1 ISO images exist
#         CD1/
#         CD2/
#     sp2-9/        // if the 3 SP2 ISO images exist
#         CD1/
#         CD2/
#         CD3/
#     sp3-9/        // if the 3 SP3 ISO images exist
#         CD1/
#         CD2/
#         CD3/
#     yast/
#         instorder
#         order
#+--------------------------------------------------------------------------+
 {
  echo "Making the directory structure ..."
  if [ -d $rdir ]; then # the subdirectory already exists - don't overwrite it
    echo "Error: root directory $rdir exists - must be removed first (rm -fr $rdir)"
    exit
  fi
  if [ ! -d $rdir/$sles/CD1 ]; then mkdir -p $rdir/$sles/CD1; fi
  if [ ! -d $rdir/$core/CD1 ]; then mkdir -p $rdir/$core/CD1; fi
  if [ ! -d $rdir/$core/CD2 ]; then mkdir -p $rdir/$core/CD2; fi
  if [ ! -d $rdir/$core/CD3 ]; then mkdir -p $rdir/$core/CD3; fi
  if [ ! -d $rdir/$core/CD4 ]; then mkdir -p $rdir/$core/CD4; fi
  if [ ! -d $rdir/$core/CD5 ]; then mkdir -p $rdir/$core/CD5; fi
  if [ ! -d $rdir/yast ]; then mkdir -p $rdir/yast; fi
  if [ $1 = 1 ]; then # SP1 ISO images exist
    if [ ! -d $rdir/$sp1/CD1 ]; then mkdir -p $rdir/$sp1/CD1; fi
    if [ ! -d $rdir/$sp1/CD2 ]; then mkdir -p $rdir/$sp1/CD2; fi
  elif [ $1 = 2 ]; then # SP2 ISO images exist
    if [ ! -d $rdir/$sp2/CD1 ]; then mkdir -p $rdir/$sp2/CD1; fi
    if [ ! -d $rdir/$sp2/CD2 ]; then mkdir -p $rdir/$sp2/CD2; fi
    if [ ! -d $rdir/$sp2/CD3 ]; then mkdir -p $rdir/$sp2/CD3; fi
  elif [ $1 = 3 ]; then # SP3 ISO images exist
    if [ ! -d $rdir/$sp3/CD1 ]; then mkdir -p $rdir/$sp3/CD1; fi
    if [ ! -d $rdir/$sp3/CD2 ]; then mkdir -p $rdir/$sp3/CD2; fi
    if [ ! -d $rdir/$sp3/CD3 ]; then mkdir -p $rdir/$sp3/CD3; fi
  fi
 }

#+--------------------------------------------------------------------------+
function mk_symbolic_links
# Make symbolic links as specified in the SLES9 install manual
#   arg 1: service pack level (0 = vanilla)
#+--------------------------------------------------------------------------+
 {
  echo "Making symbolic links ..."
  cd $rdir
  ln -fs $sles/CD1/boot boot
  ln -fs $sles/CD1/content content
  ln -fs $sles/CD1/control.xml control.xml
  ln -fs $sles/CD1/media.1 media.1
  if [ $1 = 1 ]; then # the SP1 ISO images exist
    ln -fs $sp1/CD1/driverupdate
    ln -fs $sp1/CD1/linux
  elif [ $1 = 2 ]; then # the SP2 ISO images exist
    ln -fs $sp2/CD1/driverupdate
    ln -fs $sp2/CD1/linux
  elif [ $1 = 3 ]; then # the SP3 ISO images exist
    ln -fs $sp3/CD1/driverupdate
    ln -fs $sp3/CD1/linux
  fi
  cd ..
 }

#+--------------------------------------------------------------------------+
function clean_mounts
# Clean up loopback mount points
#   arg 1: service pack level (0 = vanilla)
#+--------------------------------------------------------------------------+
 {
  echo "Cleaning up mount points ..."
  umount $rdir/$sles/CD1/
  umount $rdir/$core/CD1/
  umount $rdir/$core/CD2/
  umount $rdir/$core/CD3/
  umount $rdir/$core/CD4/
  umount $rdir/$core/CD5/
# check for Service Packs
  if [ $1 = 1 ]; then # then SP1 ISO images exist
    umount $rdir/$sp1/CD1/
    umount $rdir/$sp1/CD2/
  elif [ $1 = 2 ]; then # then SP2 ISO images exist
    umount $rdir/$sp2/CD1/
    umount $rdir/$sp2/CD2/
    umount $rdir/$sp2/CD3/
  elif [ $1 = 3 ]; then # then SP3 ISO images exist
    umount $rdir/$sp3/CD1/
    umount $rdir/$sp3/CD2/
    umount $rdir/$sp3/CD3/
  fi
  echo "Cleaning up remains of $rdir ..."
  rm -fr $rdir
  echo "Cleaned up - exiting"
  echo "You may need more loopback devices"
  exit
 }

#+--------------------------------------------------------------------------+
function mount_CDs
# Check that the 6 ISO images exist then mount and copy them
#   arg 1: service pack level (0 = vanilla)
#+--------------------------------------------------------------------------+
 {
  echo "Mounting ISO images loopback ..."
  echo "  Mounting $rdir/$sles/CD1/ ..."
  mount -o loop,ro $cd1 $rdir/$sles/CD1/
  if [ $? -ne 0 ]; then clean_mounts $1; fi
  echo "  Mounting $rdir/$core/CD1/ ..."
  mount -o loop,ro $cd2 $rdir/$core/CD1/
  if [ $? -ne 0 ]; then clean_mounts $1; fi
  echo "  Mounting $rdir/$core/CD2/ ..."
  mount -o loop,ro $cd3 $rdir/$core/CD2/
  if [ $? -ne 0 ]; then clean_mounts $1; fi
  echo "  Mounting $rdir/$core/CD3/ ..."
  mount -o loop,ro $cd4 $rdir/$core/CD3/
  if [ $? -ne 0 ]; then clean_mounts $1; fi
  echo "  Mounting $rdir/$core/CD4/ ..."
  mount -o loop,ro $cd5 $rdir/$core/CD4/
  if [ $? -ne 0 ]; then clean_mounts $1; fi
  echo "  Mounting $rdir/$core/CD5/ ..."
  mount -o loop,ro $cd6 $rdir/$core/CD5/
  if [ $? -ne 0 ]; then clean_mounts $1; fi

# check for Service Packs
  if [ $1 = 1 ]; then # then SP1 ISO images exist
```

```
    echo "Mounting SLES9 SP1 ISO images loopback ..."
    echo "  Mounting $rdir/$sp1/CD1/ ..."
    mount -o loop,ro $sp1cd1 $rdir/$sp1/CD1
    if [ $? -ne 0 ]; then clean_mounts $1; fi
    echo "  Mounting $rdir/$sp1/CD2/ ..."
    mount -o loop,ro $sp1cd2 $rdir/$sp1/CD2
    if [ $? -ne 0 ]; then clean_mounts $1; fi
  elif [ $1 = 2 ]; then # then SP2 ISO images exist
    echo "Mounting SLES9 SP2 ISO images loopback ..."
    echo "  Mounting $rdir/$sp2/CD1/ ..."
    mount -o loop,ro $sp2cd1 $rdir/$sp2/CD1
    if [ $? -ne 0 ]; then clean_mounts $1; fi
    echo "  Mounting $rdir/$sp2/CD2/ ..."
    mount -o loop,ro $sp2cd2 $rdir/$sp2/CD2
    if [ $? -ne 0 ]; then clean_mounts $1; fi
    echo "  Mounting $rdir/$sp2/CD3/ ..."
    mount -o loop,ro $sp2cd3 $rdir/$sp2/CD3
    if [ $? -ne 0 ]; then clean_mounts $1; fi
  elif [ $1 = 3 ]; then # then SP3 ISO images exist
    echo "Mounting SLES9 SP3 ISO images loopback ..."
    echo "  Mounting $rdir/$sp3/CD1/ ..."
    mount -o loop,ro $sp3cd1 $rdir/$sp3/CD1
    if [ $? -ne 0 ]; then clean_mounts $1; fi
    echo "  Mounting $rdir/$sp3/CD2/ ..."
    mount -o loop,ro $sp3cd2 $rdir/$sp3/CD2
    if [ $? -ne 0 ]; then clean_mounts $1; fi
    echo "  Mounting $rdir/$sp3/CD3/ ..."
    mount -o loop,ro $sp3cd3 $rdir/$sp3/CD3
    if [ $? -ne 0 ]; then clean_mounts $1; fi
  fi
 }

#+--------------------------------------------------------------------------+
function mk_order_files
# Create yast/instorder and yast/order files per SLES Install manual
#   arg 1: service pack level (0 = vanilla)
#+--------------------------------------------------------------------------+
 {
  echo "Creating yast/*order files ..."
  cd $rdir
  if [ $1 = 0 ]; then # the SP1 ISO images exist - add 3 entries
    echo "  Creating files for SLES9..."
    echo -e "/$sles/CD1 /$sles/CD1" > yast/instorder
    echo -e "/$core/CD1 /$core/CD1" >> yast/instorder
  elif [ $1 = 1 ]; then # the SP1 ISO images do not exist - add 2 entries
    echo "  Creating files for SLES9 + SP1..."
    echo -e "/$sp1/CD1 /$sp1/CD1" > yast/instorder
    echo -e "/$sles/CD1 /$sles/CD1" >> yast/instorder
    echo -e "/$core/CD1 /$core/CD1" >> yast/instorder
  elif [ $1 = 2 ]; then # the SP2 ISO images do not exist - add 2 entries
    echo "  Creating files for SLES9 + SP2..."
    echo -e "/$sp2/CD1 /$sp2/CD1" > yast/instorder
    echo -e "/$sles/CD1 /$sles/CD1" >> yast/instorder
    echo -e "/$core/CD1 /$core/CD1" >> yast/instorder
  elif [ $1 = 3 ]; then # the SP3 ISO images do not exist - add 2 entries
    echo "  Creating files for SLES9 + SP3..."
    echo -e "/$sp3/CD1 /$sp3/CD1" > yast/instorder
    echo -e "/$sles/CD1 /$sles/CD1" >> yast/instorder
    echo -e "/$core/CD1 /$core/CD1" >> yast/instorder
  fi
  cp yast/instorder yast/order
  cd ..
 }

#+--------------------------------------------------------------------------+
function merge_patches
# Merge patches from $rdir/sp-<n>/CD[1-3]/s390[x]/update/SUSE_CORE/9 to $rdir
#   arg 1: service pack level
#+--------------------------------------------------------------------------+
 {
  echo "Merging patches for service pack $1 ..."
  cd $rdir
  if [ $1 = 1 ]; then # merge service pack 1
    for i in 1 2; do
      rsync -HlogptrS $sp1/CD$i/$arch .
      rm -fr $sp1/CD$i/$arch
    done
  elif [ $1 = 2 ]; then # merge service pack 2
    for i in 1 2; do
      rsync -HlogptrS $sp2/CD$i/$arch .
      if [ $mounts = "no" ]; then rm -fr $sp2/CD$i/$arch; fi
    done
    echo "Hacking mediamap file for service pack $1 ..."
    cd $arch/update/SUSE-CORE/9/patches
    cp mediamap mediamap.orig
    sed -e 's/ 2/ 1/g' mediamap.orig > mediamap
    cd ../../../../..
  elif [ $1 = 3 ]; then # merge service pack 3
    for i in 1 2; do
      rsync -HlogptrS $sp3/CD$i/$arch .
      if [ $mounts = "no" ]; then rm -fr $sp3/CD$i/$arch; fi
    done
    echo "Hacking mediamap file for service pack $1 ..."
    cd $arch/update/SUSE-CORE/9/patches
    cp mediamap mediamap.orig
    sed -e 's/ 2/ 1/g' mediamap.orig > mediamap
    cd ../../../../..
  fi
  cd $arch/update
  ln -fs SUSE-CORE SUSE-SLES
  cd ..
 }

#+--------------------------------------------------------------------------+
function process_args
#
#   args: all arguments passed to the script - valid values are:
#     s390     31-bit architecture
#     s390x    64-bit architecture
#     -m       use loopback mounts for CDs, don't copy
#     --mounts use loopback mounts for CDs, don't copy
#+--------------------------------------------------------------------------+
 {
  while (( "$#" )); do
    case $1 in
      "s390")
        arch=$1
        ;;
      "s390x")
        arch=$1
        ;;
      "-m")
        mounts="yes"
        ;;
      "--mounts")
        mounts="yes"
        ;;
      *)
        echo "Error: Unrecognized parameter: $1"
        usage
    esac
    shift
  done # while there are more parms
 }

# main() - mainline code starts here - modify file names below if necessary
mounts="no"
arch="not set"
process_args $@
if [[ $arch = "s390" ]]; then # set the SLES9 s390 .iso file names
# hard-coded file names for the 6 31-bit SLES9 CD .iso images
  cd1="SLES-9-s390-RC5a-CD1.iso"
  cd2="SLES-9-s390-RC5-CD2.iso"
  cd3="SLES-9-s390-RC5-CD3.iso"
  cd4="SLES-9-s390-RC5-CD4.iso"
  cd5="SLES-9-s390-RC5-CD5.iso"
  cd6="SLES-9-s390-RC5-CD6.iso"
# hard-coded file names for the 2 31-bit SLES9 Service Pack 1 CD .iso images
  sp1cd1="SLES-9-SP-1-s390-RC5-CD1.iso"
  sp1cd2="SLES-9-SP-1-s390-RC5-CD2.iso"
# hard-coded file names for the 3 31-bit SLES9 Service Pack 2 CD .iso images
  sp2cd1="SLES-9-SP-2-s390-GM-CD1.iso"
  sp2cd2="SLES-9-SP-2-s390-GM-CD2.iso"
  sp2cd3="SLES-9-SP-2-s390-GM-CD3.iso"
# hard-coded file names for the 3 31-bit SLES9 Service Pack 3 CD .iso images
  sp3cd1="SLES-9-SP-3-s390-GM-CD1.iso"
  sp3cd2="SLES-9-SP-3-s390-GM-CD2.iso"
  sp3cd3="SLES-9-SP-3-s390-GM-CD3.iso"
elif [[ $arch = "s390x" ]]; then # set the SLES9 s390x .iso file names
# hard-coded file names for the 6 64-bit SLES9 CD .iso images
  cd1="SLES-9-s390x-RC5a-CD1.iso"
  cd2="SLES-9-s390x-RC5-CD2.iso"
  cd3="SLES-9-s390x-RC5-CD3.iso"
  cd4="SLES-9-s390x-RC5-CD4.iso"
  cd5="SLES-9-s390x-RC5-CD5.iso"
  cd6="SLES-9-s390x-RC5-CD6.iso"
# hard-coded file names for the 2 64-bit SLES9 Service Pack 1 CD .iso images
  sp1cd1="SLES-9-SP-1-s390x-RC5-CD1.iso"
  sp1cd2="SLES-9-SP-1-s390x-RC5-CD2.iso"
# hard-coded file names for the 3 64-bit SLES9 Service Pack 2 CD .iso images
  sp2cd1="SLES-9-SP-2-s390x-GM-CD1.iso"
  sp2cd2="SLES-9-SP-2-s390x-GM-CD2.iso"
  sp2cd3="SLES-9-SP-2-s390x-GM-CD3.iso"
# hard-coded file names for the 3 31-bit SLES9 Service Pack 3 CD .iso images
  sp3cd1="SLES-9-SP-3-s390x-GM-CD1.iso"
  sp3cd2="SLES-9-SP-3-s390x-GM-CD2.iso"
  sp3cd3="SLES-9-SP-3-s390x-GM-CD3.iso"
else # the one required parameter is not correct - give help
  echo "Error: Missing parameter 's390' or 's390x'"
  usage
```

```
fi
# set the directory names that will be created beneath the root
sles="sles9"
core="core9"
sp1="sp1-9"
sp2="sp2-9"
sp3="sp3-9"

echo "Making a SLES9 install tree ..."
check_for_discs
spLevel=$?
if [ $arch = "s390" ]; then
  rdir="sles9"
else
  rdir="sles9x"
fi
if [ $spLevel = 0 ]; then
  rdir="$rdir""root"  # the subdirectory in which the SLES9 install tree will be built
  echo "The tree named $rdir/ will be SLES9 without any service packs ..."
elif [ $spLevel = 1 ]; then
  rdir="$rdir""sp1root"  # the subdirectory in which the SLES9 install tree will be
built
  echo "The tree named $rdir/ will be SLES9 + SP1 ..."
elif [ $spLevel = 2 ]; then
  rdir="$rdir""sp2root"  # the subdirectory in which the SLES9 install tree will be
built
  echo "The tree named $rdir/ will be SLES9 + SP2 ..."
elif [ $spLevel = 3 ]; then
  rdir="$rdir""sp3root"  # the subdirectory in which the SLES9 install tree will be
built
  echo "The tree named $rdir/ will be SLES9 + SP3 ..."
fi
mk_directory_structure $spLevel
if [ $mounts = "no" ]; then
  copy_CDs $spLevel
else # use loopback mounts
  mount_CDs $spLevel
fi
mk_symbolic_links $spLevel
mk_order_files $spLevel
if [ $spLevel -gt 0 ]; then # merge the patches for the SP into
$rdir/s390x/update/SUSE-CORE
  merge_patches $spLevel
fi
echo "The install tree is built under $rdir/"
```

A.6.2 The clone.sh script

This section lists the **`clone.sh`** script. See section 9.2, “Cloning one new virtual server” on page 134 for a description:

Following are the possible return codes:

0	Success
1	Missing arg1 - target ID
2	Target ID not logged off
3	Target ID does not exist
4	Unexpected rc from QUERY target ID
5	191 disk not found

6	Target ID PARMFILE not found
7	User decided not to proceed
8	Can't copy target ID 102 disk
9	Can't copy target ID 100 disk
10	Error modifying networking info

Example A-6 is the source code:

Example: A-6 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"
#   on the Web at: ???
#
# ---------------------------------------------------------------------------
# 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 [-v] target_linux_ID"
  echo ""
  echo "  Clone Linux system from $master_mnt_pt to target_linux_ID"
  echo "  Option -v: verbose"
  exit 1
 }

#+--------------------------------------------------------------------------+
function process_arguments()
# give help
#+--------------------------------------------------------------------------+
 {
  verbose="off"
  if [ $# = 0 -o $# -gt 2 ]; then # user did not pass 1 or 2 arguments
    echo "Error: wrong number of arguments"
    echo ""
    help
  fi
  for arg in $*
  do
    case $arg in
      -v)
         verbose="on"
         ;;
      *)
        clone_id=`echo $arg | tr '[a-z]' '[A-Z]'`     # fold target user ID to upper
case
        ;;
    esac
  done
  if [ $clone_id = "" ]; then # new linux user ID was not passed
    echo "Error: new Linux user ID must be an argument"
    echo ""
    help
  fi
 }

#+--------------------------------------------------------------------------+
function cp_cmd()
# 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"
  ret_val=0
  ret_val=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
  return $ret_val
 }

#+--------------------------------------------------------------------------+
function check_target_id()
# Verify user ID exists and is logged off
#+--------------------------------------------------------------------------+
 {
  cp_cmd QUERY $clone_id
  case $? in
    0)  # user ID is logged on or disconnected
      echo "$clone_id user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$clone_id user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "$clone_id user ID must exist and be logged off"
      exit 4
  esac
 }

#+--------------------------------------------------------------------------+
function prepare_ipaddr()
# Prepare an IP address by adding a backslash before any "."s
# to make it 4 digits
#    Arg 1: The IP address to be modified
# Return:
#    The new value is written to the global variable new_ipaddr
```

```
#+--------------------------------------------------------------------------+
 {
  new_ipaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }

#+--------------------------------------------------------------------------+
function prepare_vaddr()
# Prepare a z/VM virtual 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 new_vaddr
#+--------------------------------------------------------------------------+
 {
  new_vaddr=`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 $leadingZeros ]; do
    new_vaddr="0$new_vaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function copy_disk()
# Try to use z/VM FLASHCOPY to copy one disk to another.
#   Arg 1: Source minidisk vaddr
#   Arg 2: the word "to"
#   Arg 3: Target user ID
#   Arg 4: Target virtual address
# Return code
#   0: success with FLASHCOPY
#   1: success with dasdfmt and dd
#   2: Target vaddr already in use?
#   3: CP LINK failed
#   4: unable to enable target minidisk
#   5: can't find source disk in /dev/dasd/devices
#   6: can't find target disk in /dev/dasd/devices
#   7: dasdfmt failed
#   8: dd failed
#+--------------------------------------------------------------------------+
 {
  ret_val=0
  source_mdisk=$1
  target_userid=$3
  target_vaddr=$4
  cp_cmd QUERY VIRTUAL $target_vaddr
  rc=$?
  if [ $rc != 40 ]; then
    echo "Error: CP QUERY VIRTUAL $target_vaddr failed with $rc"
    return 2
  fi
  cp_cmd LINK $target_userid $source_mdisk $target_vaddr MR
  rc=$?
  if [ $rc != 0 ]; then # LINK failed
    echo "Error: CP LINK $target_userid $source_mdisk $target_vaddr failed with $rc"
    return 3
  fi
  cp_cmd FLASHCOPY $source_mdisk 0 END $target_vaddr 0 END
  rc=$?
  if [ $rc != 0 ]; then # FLASHCOPY failed
    echo "FLASHCOPY $source_mdisk $target_vaddr failed with $rc"
    echo "Falling back to dasdfmt and dd copy"
#   enable target disk
    sleep 1
    chccwdev -e 0.0.$target_vaddr
    rc=$?
    if [ $rc != 0 ]; then # unable to enable target disk
      echo "Error: unable to enable 0.0.$target_disk, rc from chccwdev = $rc"
      cp_cmd DETACH $target_vaddr
      return 4
    fi
#   get device name of source disk
    source_dev=`cat /proc/dasd/devices | grep "$source_mdisk(ECKD)" | awk '{ print $7
}'`
    if [ "$source_dev" = "" ]; then
      cat /proc/dasd/devices
      echo "Error: can't find $source_mdisk(ECKD) in /proc/dasd/devices"
      chccwdev -d 0.0.$target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 5
    fi
#   get device name of target disk
    target_dev=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7
}'`
    if [ "$target_dev" = "" ]; then
      cat /proc/dasd/devices
      echo "Error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
      chccwdev -d 0.0.$target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 6
    fi
#   dasdfmt target disk
    echo "Invoking command: dasdfmt -b 4096 -y -f /dev/$target_dev"
    dasdfmt -b 4096 -y -f /dev/$target_dev
    rc=$?
    if [ $rc != 0 ]; then # dasdfmt failed
      echo "Error: dasdfmt -b 4096 -y -f /dev/$target_dev failed with $rc"
      chccwdev -d 0.0.$target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 7
    fi
    nblks=`cat /proc/dasd/devices | grep $target_dev | awk '{ print $13 }'`
#   copy from source disk to target via dd
    echo "Invoking command: dd bs=4096 count=$nblks if=/dev/$source_dev
of=/dev/$target_dev"
    dd bs=4096 count=$nblks if=/dev/$source_dev of=/dev/$target_dev
    rc=$?
    if [ $rc != 0 ]; then # dd failed
      echo "Error: dd bs=4096 count=$nblks if=/dev/$source_dev of=/dev/$target_dev
failed with $rc"
      chccwdev -d 0.0.$target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 8
    fi
    chccwdev -d 0.0.$target_vaddr
    echo "Copying disk via dasdfmt/dd succeeded ..."
    ret_val=1 # success with dasdfmt and dd
  fi # if FLASHCOPY failed
  sync # sync disks
  cp_cmd DETACH $target_vaddr
  echo "Copying disk via FLASHCOPY succeeded ..."
  return $ret_val
 } # copy_disk()

#+--------------------------------------------------------------------------+
function get_parmfile_info()
# Bring 191 minidisk online to be read by cmsfs and check for two PARMFILEs
#+--------------------------------------------------------------------------+
 {
  # recycle 191 to pick up latest changes
  chccwdev -d 0.0.0191
  chccwdev -e 0.0.0191
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "unable to enable 0.0.0191, rc from chccwdev = $rc"
    exit 5
  fi
  csmdevice=$(cat /proc/dasd/devices | grep 0191 | awk '{print $7}')
  cmsfslst -d /dev/$csmdevice | grep -i $clone_id | grep PARMFILE
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $clone_id PARMFILE not found on 191 minidisk. Exiting"
    exit 6
  fi

# get informaton about target
  export local $(cmsfscat -a -d /dev/$csmdevice $clone_id.parmfile)
  target_hostname=$IP_HOST
  target_IP=$IP_ADDR
  target_DNS=$IP_DNS
  target_GW=$IP_GATEWAY
  target_mask=$IP_NETMASK
  target_MTU=$IP_MTU
  prepare_vaddr $READ_DEVNO
  target_readdev=$new_vaddr
  prepare_vaddr $WRITE_DEVNO
  target_writedev=$new_vaddr
  prepare_vaddr $DATA_DEVNO
  target_datadev=$new_vaddr

# get information about source
  source_guestID=$(cat /proc/sysinfo | grep "VM00 Name" | awk '{print $3}')
  export local $(cmsfscat -a -d /dev/$csmdevice $source_guestID.parmfile)
  prepare_ipaddr $IP_HOST
  source_hostname=$new_ipaddr
  prepare_ipaddr $IP_ADDR
  source_IP=$new_ipaddr
  prepare_ipaddr $IP_DNS
  source_DNS=$new_ipaddr
  prepare_ipaddr $IP_GATEWAY
  source_GW=$new_ipaddr
  prepare_ipaddr $IP_NETMASK
  source_mask=$new_ipaddr
  source_MTU=$IP_MTU
  prepare_vaddr $READ_DEVNO
  source_readdev=$new_vaddr
  prepare_vaddr $WRITE_DEVNO
  source_writedev=$new_vaddr
  prepare_vaddr $DATA_DEVNO
  source_datadev=$new_vaddr
  source_domain=$(cat /etc/resolv.conf | grep domain | awk '{print $2}')
 }

#+--------------------------------------------------------------------------+
function ask_are_you_sure()
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: this will copy 100 and 102 disks to $clone_id 100 and 102"
  echo "New host name will be:  $target_hostname"
  echo "New TCP/IP address will be: $target_IP"
  echo "Other network data is retrieved from $clone_id PARMFILE on 191 disk"
  echo -n "Are you sure you want to overwrite these disks (y/n): "
  read ans
  if [ $ans != "y" ]; then
    exit 7
  fi
 }

#+--------------------------------------------------------------------------+
function copy_system()
# copy master image on 100 (root fs) and 102 (swap) to target user ID
#+--------------------------------------------------------------------------+
 {
  echo "Copying 0102 swap space to $clone_id ..."
  copy_disk 0102 to $clone_id 1102  # copy swap space
  rc=$?
  if [ $rc -gt 1 ]; then # both FLASHCOPY and dasdfmt/dd failed
    echo "Copying disk failed with $rc"
    exit 8
  elif [ $rc = 0 ]; then # FLASHCOPY succeeded
    echo "Sleeping 20 seconds for FLASHCOPY to catch up ..."
    sleep 20
  fi
  echo "Copying 0100 root file system to $clone_id ..."
  copy_disk 0100 to $clone_id 1100 # copy /sles9master to target root file system
  rc=$?
  if [ $rc -gt 1 ]; then # both FLASHCOPY and dasdfmt/dd failed
    echo "Copying disk failed with $rc"
    exit 9
  fi
 }


#+--------------------------------------------------------------------------+
function modify_cloned_image()
# Mount newly copied system over /mnt/sles9cloned and modify networking info
#   Arg 1: target userid
#   Arg 2: target minidisk
# Return code
#   0: success
#+--------------------------------------------------------------------------+
 {
  target_userid=$1
  target_mdisk=$2
  target_vaddr=$3
  echo "Mounting newly cloned image over $cloned_mnt_pt ..."
  cp_cmd LINK $target_userid $target_mdisk $target_vaddr MR
  rc=$?
  if [ $rc != 0 ]; then # LINK failed
    echo "Fatal error: CP LINK $target_userid $target_mdisk $target_vaddr MR failed with
$rc"
    exit 10
  fi
  if [ ! -d $cloned_mnt_pt ]; then
    mkdir $cloned_mnt_pt
    rc=$?
    if [ $rc != 0 ]; then
      echo "Fatal Error: mkdir $cloned_mnt_pt failed with $rc"
      exit 10
    fi
  fi
  sleep 1
  chccwdev -e 0.0.$target_vaddr
  rc=$?
  if [ $rc != 0 ]; then
    echo "Fatal error: chccwdev -e 0.0.$target_vaddr failed with $rc"
    exit 10
  fi
  cloned_DASD=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7 }'`
  if [ "$cloned_DASD" = "" ]; then
    cat /proc/dasd/devices
    echo "Fatal error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
    chccwdev -d 0.0.$target_vaddr
    exit 10
  fi
  cloned_fs="/dev/${cloned_DASD}1"
  echo "Mounting $cloned_fs over $cloned_mnt_pt ..."
  mount $cloned_fs $cloned_mnt_pt
  rc=$?
  if [ $rc != 0 ]; then
    echo "Fatal error: mount $cloned_fs $cloned_mnt_pt failed with $rc"
    exit 10
  fi
  echo "Modifying cloned image under $cloned_mnt_pt ..."
  if [ $verbose = "on" ]; then set -vx; fi
  cat $master_mnt_pt/etc/HOSTNAME | sed \
    -e "s/$source_hostname/$target_hostname/g" > $cloned_mnt_pt/etc/HOSTNAME
  cat $master_mnt_pt/etc/hosts | sed \
    -e "s/$source_IP/$target_IP/g" \
    -e "s/$source_guestID/$clone_id/g" \
    -e "s/$source_domain/$source_domain/g" > $cloned_mnt_pt/etc/hosts
  cat $master_mnt_pt/etc/defaultdomain | sed \
    -e "s/$source_domain/$source_domain/g" > $cloned_mnt_pt/etc/defaultdomain
  cat $master_mnt_pt/etc/sysconfig/network/routes | sed \
    -e "s/$source_GW/$target_GW/g" > $cloned_mnt_pt/etc/sysconfig/network/routes
  cat $master_mnt_pt/etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.$source_readdev | sed
\
    -e "s/$source_IP/$target_IP/g" \
    -e "s/$source_MTU/$target_MTU/g" \
    -e "s/$source_mask/$target_mask/g" > \
    $cloned_mnt_pt/etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.$target_readdev
  cat $master_mnt_pt/etc/resolv.conf | sed \
    -e "s/$source_domain/$source_domain/g"\
    -e "s/$source_DNS/$target_DNS/g" > $cloned_mnt_pt/etc/resolv.conf
  if [ $verbose = "on" ]; then set +vx; fi
# Regenerate SSH keys
  echo "Regenerating SSH keys in $cloned_mnt_pt/etc/ssh/ ..."
  rm $cloned_mnt_pt/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_key
# Remove any old entry, then copy clone's public key to known_hosts file
  echo "Adding $target_IP to known_hosts file"
  cd /root/.ssh
  grep -v $target_IP known_hosts > known_hosts.temp
  mv known_hosts.temp known_hosts
# construct known_hosts entry with three fields: IP@ "ssh-rsa" clones_public_key
  new_key=`cat $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key.pub | awk '{print $2}'`
  echo "$target_IP ssh-rsa $new_key" >> known_hosts
# clean up
  sync # sync disks
  umount $cloned_mnt_pt
  chccwdev -d 0.0.$target_vaddr
  cp_cmd DETACH $target_vaddr
  return 0
 }

#+--------------------------------------------------------------------------+
function make_backup_dir()
# Create a directory /backup/<VM-ID>-on-<IP.addr> if it doesn't exist
# e.g. LINUX04 with IP@ 129.40.178.44 would create /backup/LINUX04-on-129.40.178.44/
#+--------------------------------------------------------------------------+
 {
  if [ -d $backup_dir ]; then # main backup directory exists
    echo "Creating a directory under $backup_dir"
    cd $backup_dir
    backup_subdir=$clone_id-on-$target_IP
    if [ -d $backup_subdir ]; then # backup subdirectory exists
      echo "$backup_dir/$backup_subdir/ already exists"
    else
      mkdir $backup_subdir
      if [ ! $? = 0 ]; then # there was an error from mkdir command
        echo "Error creating directory $backup_dir/$backup_subdir/"
      else
        echo "Created directory $backup_dir/$backup_subdir/"
      fi
    fi
  fi
 }

# main() # the first three lines are some important global variables
master_mnt_pt="/sles9master"              # set directory of master root file system
cloned_mnt_pt="/mnt/sles9cloned"          # set directory of temporary mount point
backup_dir="/backup/linux"                # set directory of Linux backups

process_arguments $@                      # process arguments passed by user
check_target_id                           # be sure user ID exists and is logged off
get_parmfile_info                         # get information from source and target
parm files
ask_are_you_sure                          # confirm that disks will be overwritten
copy_system                               # copy 100 and 102 disks to target ID
modify_cloned_image $clone_id 100 1100    # modify newly copied system
cp_cmd XAUTOLOG $clone_id                 # bring new clone to life
make_backup_dir                           # make a backup directory
echo "Successfully cloned $master_mnt_pt to $clone_id"
echo "You should be able to ping $target_IP within one minute"
```

```
exit 0
```

A.6.3 The xipinit script

Example A-7 lists the **`xipinit`** script. See section 11.2, “Creating a DCSS/XIP2 shared file system” on page 175 for a description:

Example: A-7 xipinit script

```
#!/bin/sh
#
# /sbin/xipinit: use read only files from another file system
#
##########################################################################
#########  Change RODIRS to add the directories you are sharing.  ########
#########  Make sure you end RODIRS in a comma.                   ########
#########  Change ROMOUNT to the mount point of your choice.      ########
#########  Change DCSSNAME for the name of your DCSS.             ########
##########################################################################
RODIRS=/lib,/usr/lib,/bin,/usr/bin,/sbin,/usr/sbin,
ROMOUNT=/dcss
DCSSNAME=BC05DCSS
########
mount -t xip2 -o ro,memarea=$DCSSNAME none $ROMOUNT
# bind mount all ro dirs into rw filesystem
while [ -n "$RODIRS" ] ; do
  dir="${RODIRS%%,*}"
  RODIRS="${RODIRS#*,}"
  test -d "$dir" || continue
  echo "binding directory" $dir
  mount --bind "$ROMOUNT/$dir" "$dir"
done
# run real init
exec /sbin/init "$@"
```

A.7 Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors

A.7.1 XEDIT cheat sheet

XEDIT has line commands (Example A-8) which are typed on the command line (===>) and prefix commands (Example A-9) which are typed over the line numbers on the left side of the screen.

Example: A-8 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
```

Example: A-9 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
```

A.7.2 vi cheat sheet

Following is a small subset of vi commands, but those most commonly used.The vi editor has three modes:

1. Input mode - the **Insert** key, **i**, **o** (add a line below), **O** (add a line above) and other commands put you in this mode. 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. These commands in Example A-10 are available.

Example: A-10 vi commands

```
i      brings you back to input mode
dd     deletes a line and puts it in the buffer
<n>dd  delete <n> lines
x      delete a character
dw     delete a word
p      add the buffer past the current location
P      add the buffer before the current location
o      add a line and go into insert mode
/string - search for string
n      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, by pressing the colon (**`:`**) key, brings you to this mode. These command in Example A-11.

Example: A-11 Commands from the command line

```
:wq      save (write & quit)
:q!      quit and discard changes
:<nn>    go to line number <nn>
:r <file> read <file> into the current file
:1,$s/old/new/g globally replace <old> with <new>
:help    give help
```

Index

IBM Redbooks

z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9

(0.2”spine)
0.17”<->0.473”
90<->249 pages

z/VM and Linux on IBM System z:

The Virtualization Cookbook for SLES9

A cookbook for installing z/VM and Linux on the mainframe

Running Linux servers under z/VM made simple

Updated for z/VM 5.2 and SLES9 SP3

This IBM Redbook describes how to setup your own Linux virtual servers on IBM zSeries and System z9 under z/VM. It adopts a cookbook format that provides a clearly documented set of procedures for installing and configuring z/VM in an LPAR and then installing and customizing Linux. You need a zSeries logical partition (LPAR) with associated resources, z/VM 5.2 media, and a Linux distribution. This book is based on SUSE Linux Enterprise Server 9 (SLES9) for zSeries and we address both 31-bit and 64-bit distributions.

In addition, there are a few associated REXX EXECs and Linux scripts to help speed up the process. These tools are not IBM products nor formally supported. However, they are informally supported. They are available on the Web.

In this book, we assume that you have a general familiarity with zSeries technology and terminology. We do not assume an in-depth understanding of z/VM and Linux. This book 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

SG24-6695-01 ISBN 0738497363