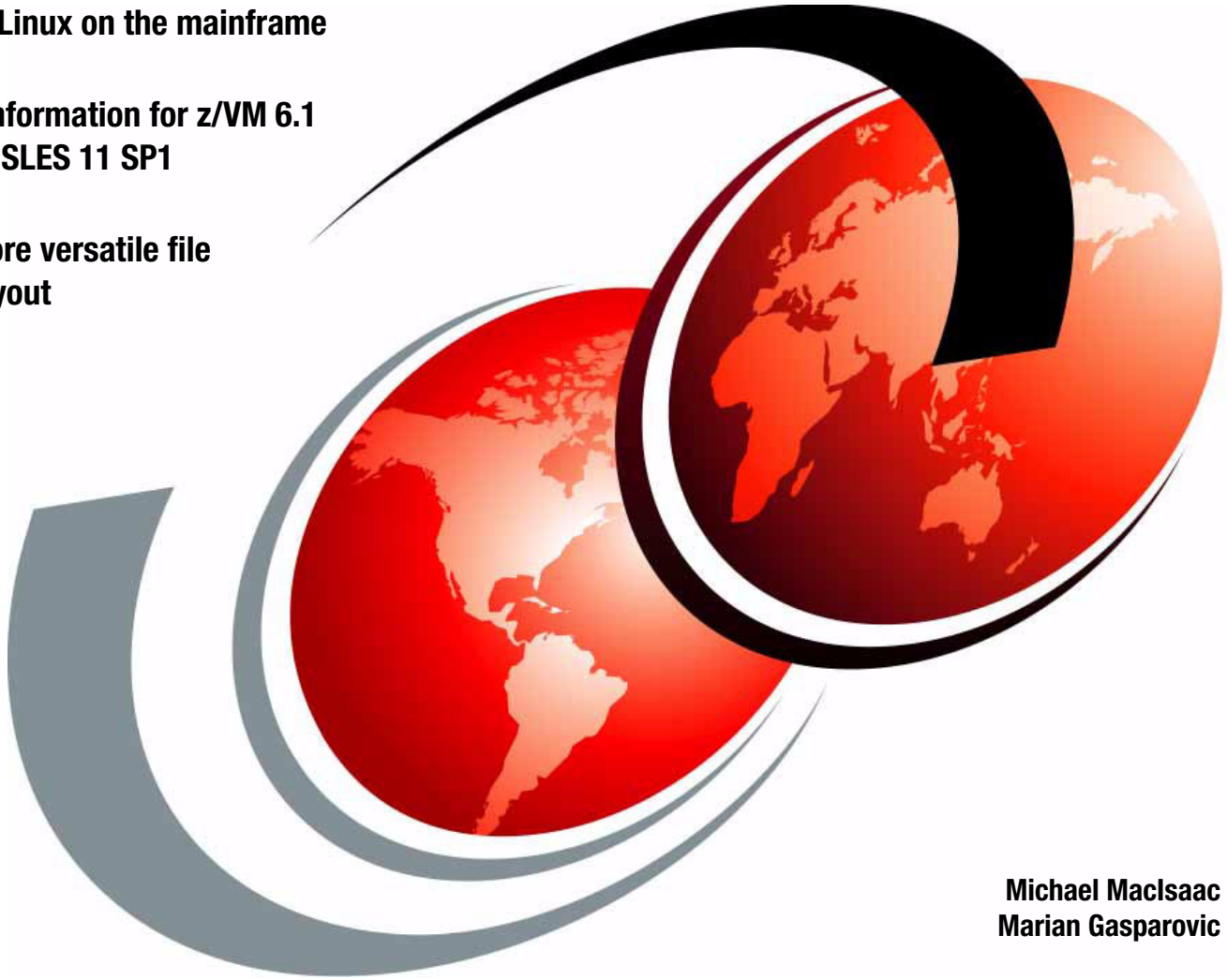
IBM

z/VM and Linux on IBM System z

The Virtualization Cookbook for SLES 11 SP1

- Hands-on instructions for installing z/VM and Linux on the mainframe
- Updated information for z/VM 6.1 and Linux SLES 11 SP1
- A new, more versatile file system layout



Michael MacIsaac
Marian Gasparovic

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1

February 2011

SG24-7931-00

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

First Edition (February 2011)

This edition applies to z/VM version 6.1 and Novell SUSE Linux Enterprise Server 11, SP1.

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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

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

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

AIX®	RACF®	System z®
DirMaint™	Redbooks®	Tivoli®
DS8000®	Redpaper™	z/OS®
ECKD™	Redbooks (logo)  ®	z/VM®
FICON®	System Storage®	z10™
IBM®	System z10®	z9®
OMEGAMON®	System z9®	zSeries®

The following terms are trademarks of other companies:

NOW, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

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

Preface

This IBM Redbooks publication describes how to create your own Linux® virtual servers on IBM® System z® hardware under z/VM®. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing and configuring z/VM in an LPAR and then installing and customizing Linux. You need an IBM System z logical partition (LPAR) with associated resources, z/VM 6.1 media, and SLES 11 SP1 Linux for System z.

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

Chapters and Appendices

The chapters and appendices in this book are as follows:

- Chapter 1, “Introduction to z/VM and Linux” on page 1 gives an overview of z/VM, gives a definition of virtualization, and describes the philosophy and choices made in the book.
- Chapter 2, “Planning” on page 7 describes how to plan hardware, software and networking resources. It discusses DASD labeling conventions used in the book and password planning. Sample worksheets are provided for the examples used in the book, as are blank copies for your use.
- Chapter 3, “Configuring a desktop machine” on page 19 describes how to set up Windows® desktops. Specifically, the following tools are discussed:
  - How to get and set up PuTTY: a commonly used SSH client
  - How to get and set up a VNC client: a tool for running graphical applications
  - 3270 emulator applications
- Chapter 4, “Installing and configuring z/VM” on page 27 shows how to install and configure z/VM. This is where you roll up your sleeves and start to work.
- Chapter 5, “Servicing z/VM” on page 73 describes how to apply service to z/VM both in the form of Programming Temporary Fixes (PTFs) and Recommended Service Upgrades (RSUs).
- Chapter 6, “Configuring an NFS server for SLES 11 SP1” on page 97, explains how to set up a temporary NFS server on a Linux PC for the purpose of installing the first Linux on System z image.
- Chapter 7, “Installing SLES 11 SP1 on the cloner” on page 105, describes how to install and configure two Linux images onto the first Linux user ID: the *golden image*, which it is cloned from, and the *cloner*, which does the cloning among other tasks.
- Chapter 8, “Installing SLES 11 SP1 on the golden image” on page 135, illustrates how to install and configrue the Linux system to be cloned and how to move the Linux install files from the Linux PC server to the cloner.
- Chapter 9, “Cloning SLES 11 SP1” on page 165 explains how to prepare z/VM user IDs and clone your first virtual server.
- Chapter 10, “Cloning open source virtual servers” on page 183, shows how to configure cloned Linux images into the following virtual servers, or *appliances*:
  - Web (Apache) virtual server
  - LDAP virtual server
  - File and print virtual server

  - Basic application development system
- Chapter 11, “Miscellaneous recipes” on page 201 describes how to add a logical volume and how to extend it if more space is needed.
- Chapter 12, “Monitoring and tuning z/VM and Linux” on page 229, describes some basic steps to begin monitoring z/VM and your new Linux virtual servers.
- Appendix A, “Important z/VM files” on page 245 provides a summary and the location of important z/VM configuration files.
- Appendix B, “z/VM and Linux code” on page 249 lists the z/VM and Linux source code associated with this book.
- Appendix C, “Additional material” on page 267 refers to additional material that can be downloaded from the Internet.
- “Related publications” on page 269 provides references to web sites, books and other pertinent information.

History

There have been quite a number of copies of *Virtualization Cookbooks*, so a short history is provided:

2010

In February of 2010 a SLES 11 book was published with the following changes:

- The z/VM sections are updated for V6.1.
- The Linux sections are updated for SLES 11.
- The REXX EXEC and XEDIT macro to help modify z/VM system labels have been removed as it is recommended to change the system labels at z/VM install time. For reference, the section on relabeling system labels is slightly modified to describe performing the steps manually, without the REXX EXEC and XEDIT macro.
- The Servicing z/VM chapter has been updated for z/VM 6.1 and now describes IBM ShopzSeries, not IBMLink.
- The Linux system that does the cloning is called the *cloner*, not the *controller.*
- A section on the X Window System has been added: see “The X Window System” on page 210.

2008

In August 2008, two IBM Redbooks® were published, one targeting Novell SLES 10 SP2 distribution and the other RHEL 5.2:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2*, SG24-7493
- *z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5.2,* SG24-7492

2007

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

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

2006

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

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

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

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

Conventions

The following font conventions are used in this book:

Monospace and bold	Commands entered by the user on the command line
MONOSPACE, bold, italics	Values in italics and bold are example values to be replaced with values correct for your enterprise
monospace	File, directories, user IDs, real and virtual device addresses

The following command conventions are used in this book:

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

The team who wrote this book

An earlier version of this book was originally written in 2005 by Michael MacIsaac, Jin Xiong and Curtis Gearhart. It was updated in 2006 by Michael MacIsaac, Carlos Ordonez and Jin Xiong. It was updated a third time in late 2006 and early 2007 by Marian Gasparovic and Mike MacIsaac.

In 2010, this book was once again updated by Mike MacIsaac and Marian Gasparovic, both of IBM, for z/VM 6.1 and Novell SLES 11 SP1.

Michael MacIsaac has been with IBM for 24 years and now works in Poughkeepsie, NY. He enjoys working on Linux and z/VM and writing IBM Redbooks publications. He currently manages a systems management development team in the z/VM organization.

Marian Gasparovic is an IT Specialist working for the IBM Server and Technology Group in IBM Slovakia. He worked as a z/OS Administrator at a Business Partner for six years. He joined IBM in 2004 as a Storage Specialist, later worked in Field Technical Sales Support for System z in the CEMAAS region as a member of a new workloads team. He joined STG Lab Services and Training recently as a System z specialist.

Sincere thanks to the following people who contributed to this project:

Sue Baloga, Bill Bitner, Carol Everitt, George Madl, Tami-Zebrowski Darrow
IBM Endicott

Roy Costa, Lydia Parziale
IBM Poughkeepsie

Dr. Manfred Gnirss, Hans-Joachim Picht, Martin Schwidefsky
IBM Boeblingen

Patrick Spinler
Mayo Clinic

Mark Post
Novell

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Directly to the authors:

  mikemac@us.ibm.com
  marian.gasparovic@sk.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Summary of changes

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

Summary of Changes
from SG24-7493-00
for z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1
as created or updated on February 22, 2011.

Summary of changes in the February 2011 version

There are both significant and smaller changes in this book:

- The Linux sections are updated for SLES 11 SP1.
- The **`yast -i`** command was replaced with **`zypper install`**.
- Sections on VM electronic download in Chapters 4 and 6.
- Chapter 5, “Servicing z/VM” on page 73 has been updated to include information on service for the new zEnterprise 196.
- Section 11.5, “Setting up Memory Hotplug” on page 215 is new.
- Section 11.6, “Utilizing the cpuplugd service” on page 218 is new.
- Section 11.7, “Hardware cryptographic support for OpenSSH” on page 223 is new.
- New document number, SG24-7931-00.

1

Introduction to z/VM and Linux

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

With a z/VM and Linux infrastructure, you can reduce the time between deciding on the acquisition of new servers and then implementing them because new servers can be deployed in a matter of minutes. With this powerful build and clone capability you can launch new products and services without the exhaustive planning, purchasing, installing and configuring new hardware and software that can be associated with conventional discrete hardware servers. Development groups who need test environments built and rebuilt rapidly to enable them to efficiently deliver their projects in good time, handling change management in the process can also benefit from this unique advantage.

Some of the mainframe's and z/VM's strengths are:

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

Much function has been added to z/VM since Version 5.2.

z/VM 5.3

z/VM 5.3 became generally available in June of 2007. Scalability was extended to allow 256 GB of real memory, a total of 8 TB of virtual storage, and 32 real processors. z/VM V5.3 also added support for the Collaborative Memory Management Assist (CMMA) on the z9®

EC and the z9 BC processors. Virtual Machine Resource Manager (VMRM) detects when memory is constrained and notifies the Linux guests, which can then adjust their memory consumption to help relieve the memory constraint. In the previous major release, z/VM 5.2, many memory contention issues were removed with the Control Program (CP) now using memory above 2 GB for a much broader set of operations. Previously, guest pages had to be moved below 2 GB for many reasons, for example in both standard I/O and Queued Direct I/O (QDIO). Now I/O can be 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-real-memory virtual server environments that are memory intensive.

z/VM 5.4

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

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

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

z/VM 6.1

z/VM 6.1, available in October of 2009, is intended to be the base for all future z/VM enhancements. This release implements a new Architecture Level Set (ALS) available only on the IBM System z10 Enterprise Class server and System z10 Business Class server and future generations of System z servers. Requiring z10 technology or later allows z/VM to take advantage of newer hardware technology for future exploitation.

Enhancements in z/VM V6.1 provide:

- Enhanced performance of virtual networking environments running heavy guest-to-guest streaming workloads
- Faster access to data when utilizing FICON® Express8
- Closer integration with IBM Systems Director to eliminate the need to download agents and help simplify the installation of those agents
- Significantly better and more highly secure guest transactions when using Crypto Express3 as compared to Crypto Express2
- Guest support for IBM System Storage® DS8000® Extended Address Volumes (EAVs) to help simplify storage management and relieve address constraints

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

http://www.vm.ibm.com

1.1 What virtualization is

Virtualization is the ability of a computer system to share resources so that one physical server can act as many *virtual servers*. z/VM allows the sharing of the mainframe's physical resources such as disk (DASD), memory (sometimes called *storage*), network adapters (OSA cards) and processor (CPs or IFLs). These resources are managed by a *hypervisor*. z/VM's hypervisor is called Control Program (CP). When you log onto z/VM, the hypervisor creates a virtual machine that can run one of many different operating systems. The two operating systems that are discussed in this book are the z/VM native one, the Conversational Monitoring System (CMS), and Linux. CMS can be thought of as a z/VM shell. Virtual machines running Linux as guests of a z/VM host become the *virtual servers*.

1.2 A philosophy adopted in this book

An important philosophy adopted in this book is to keep all solutions simple.

A lot of books and papers are writing about virtualization these days, but they rarely tell you how to do it. This book gives you the *how tos* that back up 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 be made. Here are some of the choices and assumptions made in this book:

- Use of a cloning product versus "roll your own" cloning: Cloning products, such as Aduva's Onstage, Mainstar's Provisioning Expert, IBM Tivoli® Provisioning Manager, and IBM Systems Director are outside the scope of this book. While these are all viable solutions, the cloning described in this book allows you to *roll your own* Linux images without requiring such products. However, these products are more sophisticated than the simple clone script and z/VM configuration described in this book.
- Directory Maintenance product versus the `USER DIRECT` file: The `USER DIRECT` file is chosen over a directory maintenance product such as IBM DirMaint™ or CA's VM:Direct. If you feel that DirMaint as a directory maintenance product is better for your enterprise, you can use the book *Getting Started With Linux*, SC24-6096, to configure z/VM, and 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* them (define and bring Linux user IDs online automatically) as part of the cloning process. The target Linux user ID must exist with the appropriate resources defined.
- Shared read-only file system versus read-write: Many cloning solutions use an environment that shares the `/usr/` file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. A read-write `/usr/` file system on the virtual servers is chosen to keep things as simple as possible. See:

  `http://www.vm.ibm.com/linux/dcss/`
- Conventional 3390 ECKD™ DASD versus FBA disks accessed using SCSI over FCP: The System z server has traditionally only supported 3390 DASD. Support has been extended to include SCSI/FBA disks in storage area networks (SANs). The support of FBA disks is slightly more complicated than conventional DASD. In keeping things as simple as possible, only conventional DASD is described in this book.

- Cloning script or EXEC versus manual installation: Two methods of cloning are described, manual and using a Linux bash script. The manual method is described so you can better learn the concepts. The Linux script is described and provided so you can save time.
- Use of VNC server for GUI-based tools: Some programs, such as installation tasks, require a functioning GUI. Because a GUI is only occasionally needed, it is reasonable to have a lightweight environment in place. Complete desktops such as Gnome or KDE are not just overkill for System z, they are a bad idea in a virtualized environment because they are costly in terms of resources. For this reason, a VNC server is described as the recommended GUI. See section 11.4, “The X Window System” on page 210.

1.4 Infrastructure design

To install and configure z/VM, install, configure and clone Linux, or *provision virtual servers*, there must be a certain infrastructure design in place. A System z server with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-1 shows a block diagram of a System z10 with many LPARs. z/VM 6.1 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.

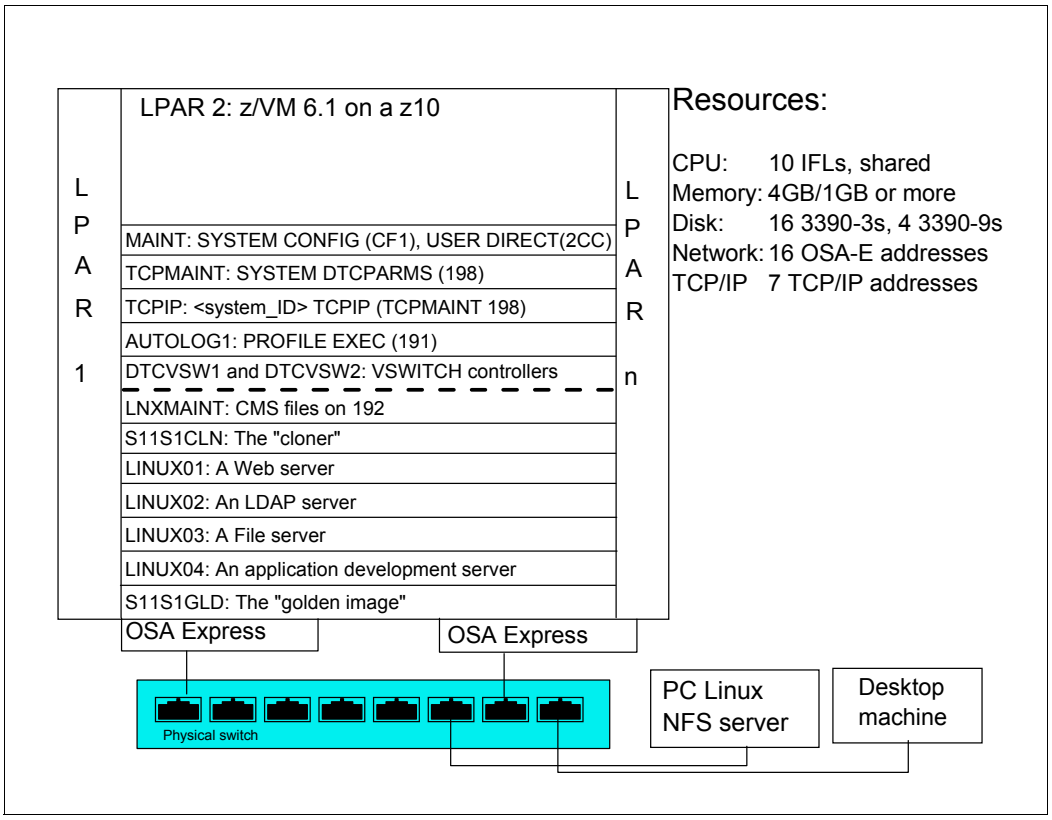


Figure 1-1 System infrastructure and z/VM user IDs

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

- LNXMAINT- A user ID on which to store files that will be used by both CMS and Linux.

- S11S1CLN- The *cloner* image that does the cloning. It also serves as the Linux install server.
- LINUX01-04- The Linux virtual servers described in the *Cloning open source virtual servers* chapter. Each virtual server is configured with two 3390-3 minidisks to allow for slightly more than 4 GB of disk space.
- S11S1GLD- The SLES 11 golden image. This is the Linux system that is cloned.

1.5 Usability tests performed for this book

During the writing of this book, many usability tests were conducted. The participants had a variety of skills, but none had both Linux and z/VM system administration skills. By the end of the first day in all of the formal tests, most participants had all completed up to and including Chapter 5, “Servicing z/VM” on page 73, so z/VM was installed, serviced and customized for TCP/IP communications with a highly available VSWITCH. By the end of the second day, most participants had cloned their first Linux virtual server. You should be able to complete most steps in the book in four solid days of work.

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 describes conventions adopted for labeling 3390 volumes. Finally, resource worksheets are presented for:

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

2.1 Bill of materials

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

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

- A System z logical partition (LPAR); on a System z10 (z/VM 6.1 will not run on earlier hardware)
  - Processors: One IFL (or CP) minimum, two or more are strongly recommended.
  - Memory: 3 GB central or 1 GB expanded minimum, 6 GB or 2 GB or more recommended. This 3:1 ratio of central to expanded storage is a good starting point for relatively small systems such as in these example sizes.

    See the following website for a discussion of how to apportion memory:

    http://www.vm.ibm.com/perf/tips/storconf.html
  - DASD: 24 3390-3s or 8 3390-9s at a minimum.

  - Open Systems Adapter (OSA) network cards: One card minimum with 8 device numbers (technically 6, but OSA “triplets” usually start on an even address). Two OSA Express cards with 8 device numbers on one and 4 on the other is recommended for high availability.
- A network-attached computer that will act as a Network File System (NFS) server temporarily with at least 6 GB of disk space, but more may be needed. Setting up a Linux PC or UNIX® server is described.
- A workstation or desktop that has network access to the mainframe.

2.1.2 Software resources

The following software resources are needed:

- z/VM 6.1 install media with documentation (installation from DVD is described in this book). There are three possible ways to get z/VM:
  - On DVD by ordering “z/VM System Image DVD” on 3590 or 3592 (you will not receive a tape, just a DVD).
  - Electronically, by ordering “z/VM 3390 System DDR” using Internet delivery (do not be fooled by the name—it is actually a soft copy of the DVD contents, not an ISO file).
  - On tape by ordering “z/VM 3390 System DDR” on 3590 or 3592.

    If you want to install from FTP, order the DDR with electronic delivery and follow the special instructions included with the order. These instructions have you load the soft copy files into your FTP server. You can also create a DVD from them for backup.

    http://www.vm.ibm.com/install/vm61inst.pdf
- SLES 11 SP1 install media or ISO image(s). See:

  http://www.novell.com/products/server/eval.html
- An operating system for the NFS server, SLES 11 SP1 is recommended.
- The code associated with this book

  ftp://www.redbooks.ibm.com/redbooks/SG247931
- Tools on the workstation and desktop:
  - A 3270 Emulator such as Attachmate Extra, Hummingbird Host Explorer, or IBM Personal Communications for Windows desktops
  - A Linux SSH client such as PuTTY (recommended) or TeraTerm
  - A VNC viewer

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

2.1.3 Networking resources

The following network resources are needed:

- A TCP/IP address for z/VM
- One TCP/IP address for each Linux virtual server
- Associated TCP/IP information:
  - DNS host name
  - DNS domain
  - DNS server TCP/IP address

- TCP/IP gateway
- TCP/IP subnet mask
- TCP/IP broadcast address (usually calculated from address and subnet mask)
- TCP/IP MTU size

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

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 System z DASD is addressed with a device number consisting of four hexadecimal digits. Each System z DASD has a 6-character label. It is convenient to include the 4-digit address in the label so that you can easily tell the address of each DASD from its label. When followed, this convention guarantees that no two DASDs will have the same label. This can be an important issue, especially when z/OS® has access to the DASD.

Sometimes DASD is shared among LPARs, in which case your z/VM LPAR can *see* DASD *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 using the first character. That leaves the second character in the label to identify the basic function of the DASD.

The LPAR used in this book is identified by the character *M*. The following characters are used for the types of DASD in the second character of the label:

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

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

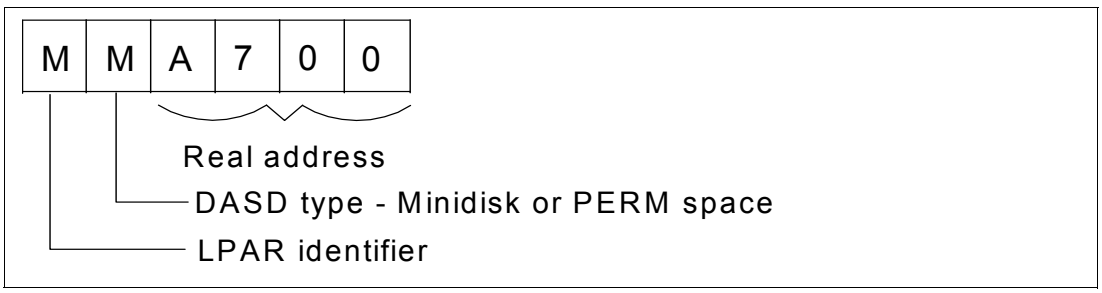


Figure 2-1 DASD labeling convention

The letter T is hard-coded into REXX EXECs that adopt this convention. If you want a different LPAR identifier character, it can easily be changed (set the `firstChar` variable).

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 file `USER DIREORIG` before it is modified the first time.

2.2.3 The command retrieve convention

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

There is a convention in z/VM to use the F12 function key (previously labeled `PF12` on physical 3270 devices) to retrieve the last command, though it is not defined to all user IDs. There is no convention to retrieve commands in the other direction but it is possible to set another key to that function. Therefore, F11 is used to *retrieve forward* 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 Disk planning

There are different aspects to consider when planning how to choose and allocate disk storage. Some aspects include the following

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

DASD vs. SCSI/FCP

This book describes how to use conventional ECKD DASD; it does not discuss FBA disks accessed over SCSI/FCP. This is not because either technology is superior, but simply because DASD seems to be much more common than SCSI/FCP disks. If you were to use SCSI/FCP disks, cloning using the **`clone.sh`** script would have to be modified to account for World Wide Port Names and Numbers. Sometimes a combination of these two types of disk storage is used. When that is the case, the ECKD emulated DASD is often used for the root file system and SCSI/FCP disks are used for large data storage areas.

3390-3s vs. 3390-9s

Emulated 3390-3s format to about 2.3 GB, while 3390-9s are three times the size or about 6.8 GB. Either size will work, though 3390-3s have been recommended over 3390-9s by some performance analysts. This book describes using a mix of 3390-3s and 3390-9s.

Disk storage per Linux image

Disk storage has the following characteristics:

- The root file system is on `/dev/dasda1` with a recommended size of 384 MB. It is not recommended that your root file system be part of a logical volume because if there are any problems with LVM, the system will not be able to boot.
- Other file systems are assigned logical volumes that are part of a single volume group with the characteristics shown in Table 2-1.

Table 2-1 Recommended logical volume file systems and sizes

Mount point	Logical volume name	Size
`/usr/`	`usr-lv`	2.5 GB
`/var/`	`var-lv`	512 MB
`/opt/`	`opt-lv`	384 MB
`/tmp/`	`tmp-lv`	384 MB

This layout uses about 4 GB out of 4.5 GB of disk space, leaving half a gigabyte free for growth (but you can always add more physical volumes to the volume group). You could also choose physical volumes with disk sizes other than 3338 cylinders. For example, if you choose to use 3390-9s, you could give `100` and `101` each half of the volume, giving each Linux about 6.8 GB of disk space.

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

2.4 Memory planning

Planning memory may be the most difficult issue with z/VM and Linux on System z, yet the most important to ensure adequate performance. The simplest solution may appear to involve having enough central memory (storage) in the LPAR so that z/VM never pages and Linux never swaps. However, such resource is often not realistically available. A good rule of thumb is to allocate memory on a just-enough basis for each Linux server. A good starting point is to set a virtual machine size by changing the memory allocation value at just over the value at which the guest starts to swap at the Linux system level when under normal loading. If some level of sustained swapping is inevitable due to the nature of the workloads, then ensure virtual disks are used for the swap media.

An understanding of memory planning is recommended. Here are some resources that cover this important topic:

- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926-01, 2008, on the web at:

  `http://www.redbooks.ibm.com/redpieces/abstracts/sg246926.html?Open`
- The IBM z/VM Performance Resource pages in general, on the web at:

  `http://www.vm.ibm.com/perf/`
- The IBM z/VM page specifically discussing memory allocation:

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

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

2.5 Password planning

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

This book considers different system administration roles:

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

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

The method of backing up z/VM data onto the Linux cloner implies 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.

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 cloner, Linux virtual server user IDs)
- The Linux virtual server users (with or without access to 3270 sessions, with or without the root passwords)

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

2.6 Planning worksheets

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

2.6.1 z/VM resources used in this book

Table 2-2 on page 13 lists the z/VM resource values used in the examples in this book. You can use these values as a reference for completing the blank worksheets that follow.

Table 2-2 z/VM resources worksheet

Name	Value	Comment
LPAR name	LVM2	16 GB central storage or 2 GB expanded, 10 shared IFLs
CPC name	H15C	Name of CPC on which the LPAR is located
z/VM system name	POKSND61	Name to be assigned to z/VM system
TCP/IP host name	gpok249	Assigned by a network administrator; helpful to set in DNS beforehand, but not necessary
TCP/IP domain name	endicott.ibm.com	Helpful to set in DNS beforehand
TCP/IP gateway	9.60.18.129	The router to and from the local subnet
DNS server 1	9.0.2.11	Assigned by the network administrator
DNS server 2/3 (optional)	9.0.3.1	Not used
OSA device name	eth0	Name of the interface to be assigned by IPWIZARD
OSA starting device number	B420	Start of OSA *triplet* for the z/VM TCP/IP stack
TCP/IP address	9.60.18.249	The TCP/IP address of the z/VM system
Subnet mask	255.255.255.128	Assigned by network administrator
OSA device type	QDIO	Often “QDIO” for OSA/Express cards
Network type	Ethernet	Usually “Ethernet”
Port name (optional)		Not required by z/VM
Router type	None	Usually “None”
MTU size	1500	Check with network administrator
Primary OSA device number for VSWITCH	B440	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	B424	Should be on a different CHPID/OSA card

2.6.2 z/VM DASD used in this book

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

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

Real device	Label	Type	Notes
6280	610RES	CP owned - 3390-3	z/VM system residence volume
6281	UV6281	CP owned- 3390-3	z/VM spool volume 1 (label set at z/VM install)
6282	UV6282	CP owned - 3390-3	z/VM paging volume 1 (label set at z/VM install)
6283	UV6283	CP owned - 3390-3	z/VM first work volume (label set at z/VM install)

Real device	Label	Type	Notes
6284	UV6284	CP owned - 3390-3	z/VM 2nd work volume (label set at z/VM install)
6285	UV6285	CP owned - 3390-3	Paging volume 2
6286	UV6286	CP owned - 3390-3	Paging volume 3
6287	UV6287	CP owned - 3390-3	Paging volume 4
6289	UM6289	System disk - 3390-3	3390-3 for `LNXMAINT` and `S11S1CLN 101`
6290	UM6290	System disk - 3390-3	3390-3 for `S11S1CLN 100`
6293	UM6293	System disk - 3390-3	3390-3 for `S11S1CLN 102`
6294	UM6294	System disk - 3390-3	3390-3 for `S11S1CLN 103`
63A2	UM63A2	System disk - 3390-9	3390-9 for `S11S1CLN` and `S11S1GLD 100`
63A9	UM63A9	System disk - 3390-9	3390-9 for `S11S1GLD 101` and `LINUX01`
63AA	UM63AA	System disk - 3390-9	3390-9 for `LINUX02` and `LINUX03 100`
63AB	UM63AB	System disk - 3390-9,	3390-9 for `LINUX03 101` and `LINUX04`
6339	UM6339	System disk - 3390-3	3390-3 for adding logical volumes
6360	UM6360	System disk - 3390-3	3390-3 for extending logical volumes

2.6.3 Linux resources used in this book

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

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

Name	Value	Comment
TCP/IP address	9.60.18.240	
User/password	root/lnx4vm	
NFS-exported install directory	`/nfs/sles11sp1/`	Directory with DVD 1

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

Table 2-5 Linux resources used in this book

Name	Value	Comment
Linux root password	lnx4vm	
TCP/IP gateway	9.60.18.129	Obtain from network administrator
Subnet mask	255.255.255.128	Obtain from network administrator
DNS server	9.0.2.11, 9.0.3.1	Obtain from network administrator
VNC installation password	12345678	Must be 8 characters

2.6.4 Linux user IDs used in this book

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

Table 2-6 Linux user ID used in this book

Linux user ID	IP address	DNS name	Notes
S11S1GLD	9.60.18.222	gpok222.endicott.ibm.com	SLES 11 golden image
S11S1CLN	9.60.18.223	gpok223.endicott.ibm.com	The cloner
LINUX01	9.60.18.224	gpok224.endicott.ibm.com	A Web virtual server
LINUX02	9.60.18.225	gpok225.endicott.ibm.com	An LDAP virtual server
LINUX03	9.60.18.226	gpok226.endicott.ibm.com	A file and print virtual server
LINUX04	9.60.18.227	gpok227.endicott.ibm.com	An application development server

2.7 Blank worksheets

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

2.7.1 z/VM resources worksheet

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

Table 2-7 z/VM resources blank worksheet

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

2.7.2 z/VM DASD worksheet

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

Table 2-8 z/VM DASD blank worksheet

Device number	Label	Type	Notes

2.7.3 Linux resources worksheet

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

Table 2-9 Linux NFS server resources blank worksheet

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

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

Table 2-10 Linux resources blank worksheet

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

2.7.4 Linux user ID worksheet

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

Table 2-11 Linux user ID blank worksheet

Linux user ID	IP address	DNS name	Notes

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 easy to use and cryptographically secure. If you are using a Linux desktop system, an SSH client is built in. But if you are using a Windows desktop, you will need a good SSH client.

PuTTY is probably the most commonly used. You can download PuTTY from the web at:

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

To download from this page, click the **putty.exe** link for your architecture. Save the file in a directory path such as `C:\WINNT`. PuTTY is a standalone executable (no installation needed other than copying the file). You may also want to create a shortcut on your desktop or task bar.

Open PuTTY and the configuration window shown in Figure 3-4 should open. If you spend a few minutes to configure PuTTY it may pay off in time savings. The examples shown are using PuTTY Release 0.60.

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

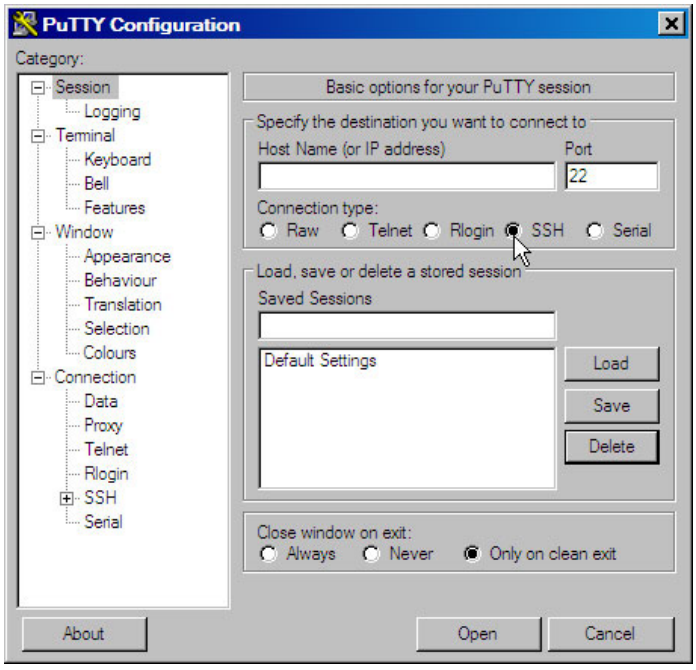


Figure 3-1 PuTTY Configuration window

3. Click **Logging** in the left panel as shown in Figure 3-2 on page 20.
   - Select **Printable output** in the **Session logging** radio group. This allows you to go back and check the output of certain commands.
   - Set the Log file name to `&H&M&D&T.log` so that a timestamp will be in the file name.

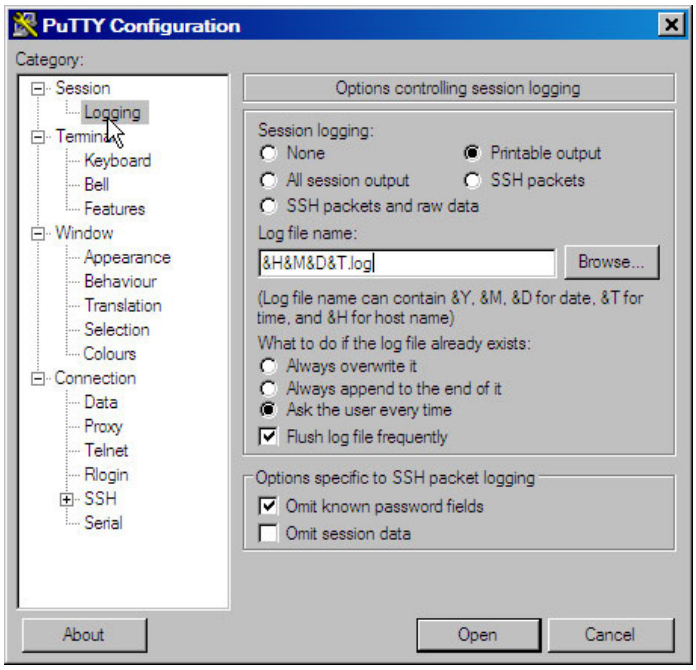


Figure 3-2 Setting logging

4. In the left panel, click **SSH** near the bottom as shown in Figure 3-3.

5. On the right side, under the **Preferred SSH protocol** version, select **2 only**.

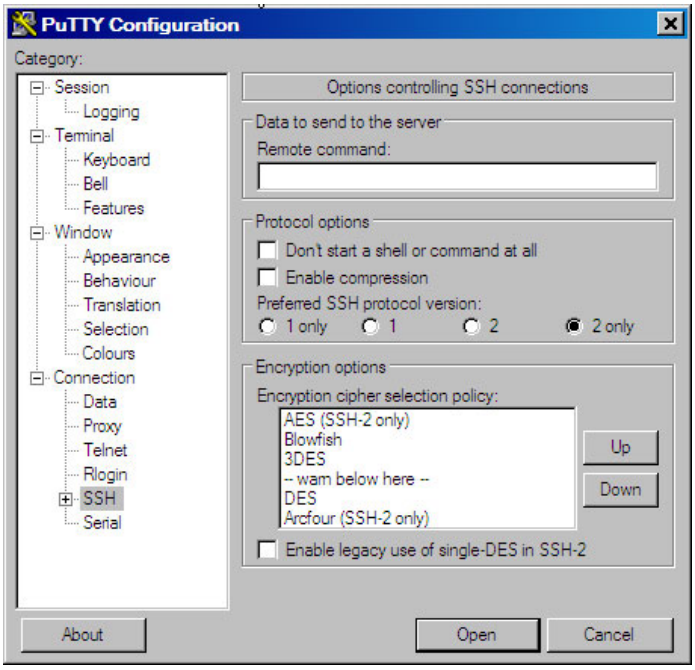


Figure 3-3 Setting SSH Protocol 2

6. In the left Category panel, click **Terminal** as shown in Figure 3-4 on page 21.
7. Select the **Use background colour to erase screen** check box, which results in a better job of painting the panel for applications that use curses (block graphics).

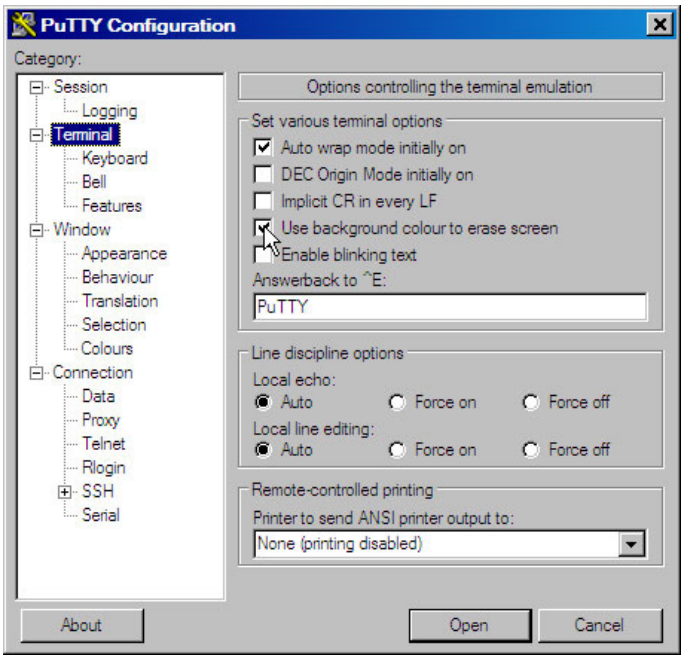


Figure 3-4 Customizing PuTTY SSH settings (Part 1 of 4)

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

9. You may choose a larger window size and more lines of scrollback. In this example, 50 rows, 100 columns and 1000 lines of scrollback are set.

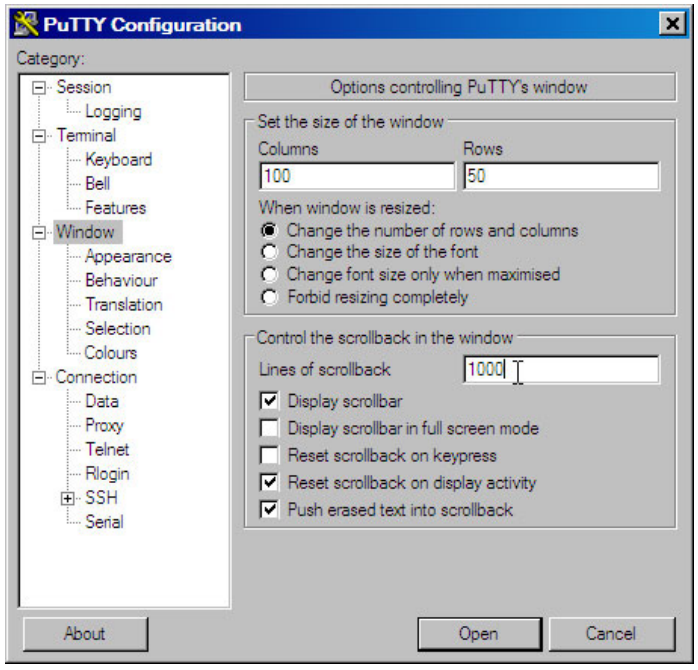


Figure 3-5 Setting Window and scrollback size (Part 2 of 4)

10. Click **Session** in the left pane as shown in Figure 3-6 on page 22.
11. Click **Default Settings** in the Saved Sessions pane, then click **Save**. This makes all future sessions that you define inherit the preferences you just set.

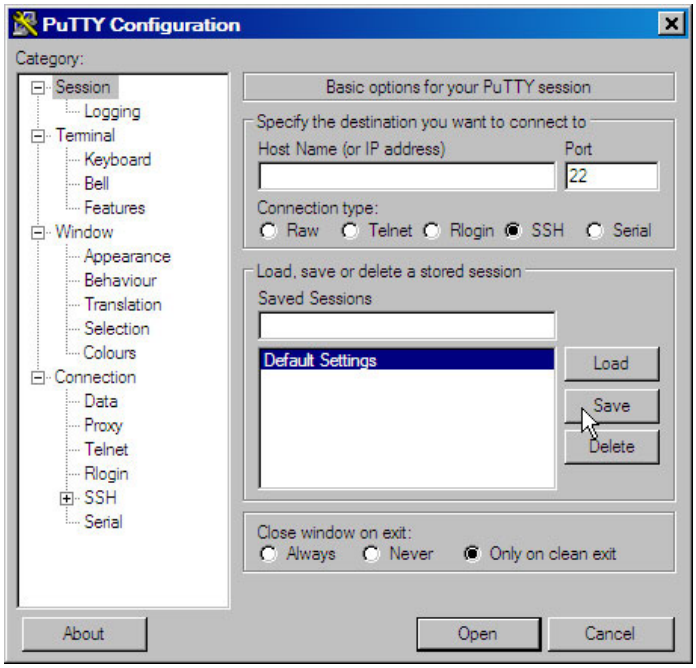


Figure 3-6 Saving new default settings (Part 3 of 4)

Saving sessions

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

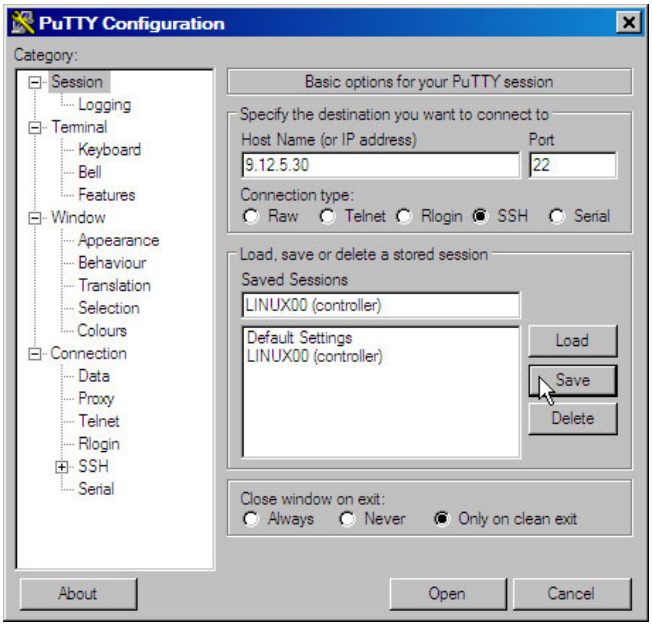


Figure 3-7 Customizing PuTTY window settings (Part 4 of 4)

Now to save a session for each virtual server, do the following:

1. In the Host Name (or IP address) field, enter the TCP/IP address (or DNS name).
2. In the Saved Sessions text area, choose a name that you will remember. In this example, the name `LINUX00 (cloner)` is used.
3. Again click **Save** and you should see the name added to the Saved Session list.

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

3.2 Setting up a VNC client

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

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

3.2.1 Downloading and running RealVNC

If you have a Windows desktop, the VNC client from RealVNC is a popular choice. You can purchase a full function RealVNC client, or there is a free version. The RealVNC home page is at:

`http://www.realvnc.com`

The download page is at:

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

Click **Download and Use**. Fill out the web form and download the executable. When you have downloaded it, run it and an install program will start. At the time of writing of this book, RealVNC 4.1.2 was the current version.

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

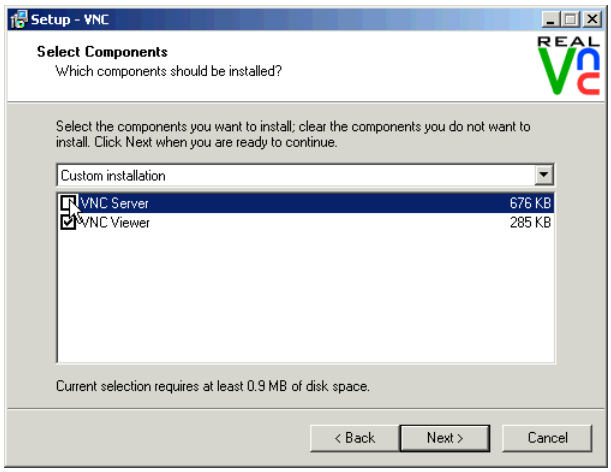


Figure 3-8 RealVNC Select Components panel

Complete the panels and the installation process should go quickly.

3.3 3270 emulators

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

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

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

- Set the **Enter** and **Clear** function keys to be where you would expect them. On some emulators, the default Enter key action is set to the right Ctrl key of modern keyboards. Likewise the Clear key action is sometimes set to the Esc key in the upper left corner of modern keyboards or the Pause key in the upper right.
- Set a larger window. Often the default number of lines in an emulator session is 24. You will probably be more productive with a 32, 43 or more lines if they can easily fit in a window given your desktop display size and resolution.
- Have the session automatically reconnect after logoff. Having a new logon panel come back immediately after you log off can also save you time in the long run. This is often not the default behavior.
- Save your connection sessions. Rather than continually typing in the IP address or DNS name of the z/VM system to which you want to connect, spend a few minutes to define and save a session for each system to which you may 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

z/VM can be installed first level from tape, from DVD, or from an FTP server. Installing from tape is not described in this book. However, installing from the physical media of DVDs, or without physical media, from an FTP server, are.

To complete this chapter, you must complete the majority of Chapter 6, “Configuring an NFS server for SLES 11 SP1” on page 97. If you are installing z/VM from an FTP server, you should complete section 4.1, “Installing z/VM from DVD or FTP server” on page 28, then complete Chapter 6.

We recommend that you start here, because there is a step when installing z/VM (instdvd) that can take two or more hours to complete. While that process is running, you can complete Chapter 6. Alternatively, if you have other personnel who can work on the project, you can start both chapters at the same time on the different systems.

This chapter consists of the following sections that should be completed:

- “Installing z/VM from DVD or FTP server” on page 28
- “Configuring TCP/IP” on page 41
- “Configuring the XEDIT profile” on page 43
- “Customizing the SYSTEM CONFIG file” on page 44
- “Configuring TCP/IP to start at IPL time” on page 46
- “Adding paging volumes” on page 50
- “Creating a user ID for common files” on page 56

In addition, there are optional sections:

- “Addressing z/VM security issues” on page 63
- “Backing up your z/VM system to tape” on page 65
- “Relabeling system volumes” on page 65
- “Restoring your z/VM system from tape” on page 71

4.1 Installing z/VM from DVD or FTP server

This section assumes a first level installation of z/VM from DVD onto 3390 DASD. If you have not already done so, complete the worksheet in 2.7.1, “z/VM resources worksheet” on page 16.

For System z9® hardware and older, you will need access to the Hardware Management Console (HMC) with a user ID that has authority to go into *single object operations mode*, though this is not pertinent for z/VM 6.1 because it installs onto System z10 or later. The requirement to be in single object operations mode for z10 or later has been removed.

z/VM 6.1 is shipped on tape, on DVD, and is available from the Internet through electronic download. z/VM should install faster from tape due to better I/O speeds; however, installing from tape is becoming less common.

If you are not familiar with the HMC and z/ VM, you may want to use the complete installation manual *z/VM Guide for Automated Installation and Service,* Version 6 Release 1.0, GC24-6097. If you are installing z/VM at the *second level* (z/VM under z/VM) or onto FCP/SCSI disk, you will want to use this z/VM manual because the sections that follow do not address these options.

4.1.1 Obtaining z/VM through electronic download

z/VM can be ordered and delivered electronically through IBM ShopzSeries. A detailed discussion is outside the scope of this book; however, short steps are documented. Note that the steps and links may change over time, but the basic process should remain the same.

You may download the z/VM product install files to a staging machine, such as a Windows desktop, as was done in this example, and later upload them to an FTP server. However, you may also download them directly to the machine that will be the FTP server, such as a Linux PC if it has access to the Internet and a browser.

To order z/VM, perform the following steps:

- Go to the z/VM service page at:

  `http://www.vm.ibm.com/service/`
- Click **IBM ShopzSeries** in the section IBM Support Portals.
- Sign in by clicking **Sign in for registered users** in the upper right.
- Click **create new software orders**.
- On *Step 1*, select **z/VM Products** and choose **VM SDO version 6** in the dropdown menu to the right. Click **Continue**.
- On *Step 2*, select a hardware system on which you plan to run z/VM from the list of Hardware systems for your customer number, and click **Continue**.
- On *Step 3*, for the Filter, select **VM - VM Base Product,** select your language and for the *Filter*, select **Show all products**, then click **Show catalog**. A submenu appears.

  Select **z/VM V6 3390 System DDR** and click **Continue**.
- On *Step 4*, verify the order and click **Continue**.
- On *Step 5*, verify the entitlements and click **Continue**.
- On *Step 6*, for the Preferred media, select **Internet** and click **Continue**.
- On *Step 7*, review and click **Submit**.

- It may take some time for the order to be prepared. In this example, the e-mail stating that the order was ready for download was received after about four hours. When you receive the e-mail, it will contain the URL for downloading your order. Use a browser to go to that URL.
- From that address, there will be links to investigate as shown in Figure 4-1. It has the following five sections:
  - Order Packing List - The list of available products and manuals
  - Installation Instructions - Clicking **View now** will take you to a web page:

    `http://www.vm.ibm.com/install/vm61inst.pdf`

    This PDF describes in general terms how to go from the product install files to physical DVDs or to an FTP server. If you want to go from the product install files to physical DVDs, you should complete this section, but will not need to use the later section on how to set up an FTP server. If you want to use an FTP server to avoid physical media altogether, you can read the PDF for a general approach, and then complete this section and 6.4, “Configuring an FTP server for z/VM installation” on page 101 for specific details.
  - Product Publications - Allows you to access z/VM publications related to installation.
  - Additional Publications - Allows you to download a z/VM SDO document (4 pages).
  - VM product material - This is the most important section because it is where you go to download z/VM product installation files. In the example used in this book, the link **Download to your workstation using IBM Download Director** was clicked as shown in Figure 4-1.

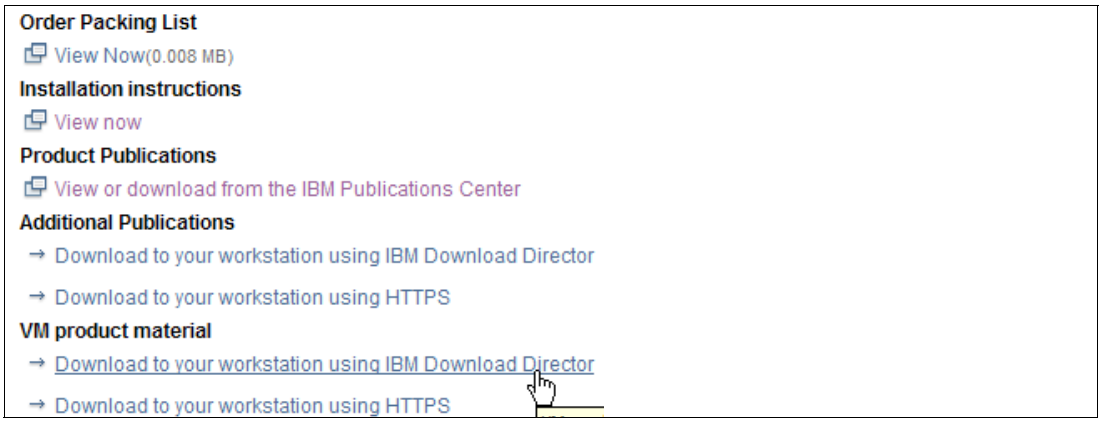


Figure 4-1 Web page for downloading z/VM electronically

- Clicking this link brought up the panel shown in Figure 4-2 on page 30. The first and third check boxes were selected as z/VM is being installed onto 3390 DASD. The 1.3 GB of data was downloaded relatively quickly due to multiple connections being opened through the use of IBM Download Director.

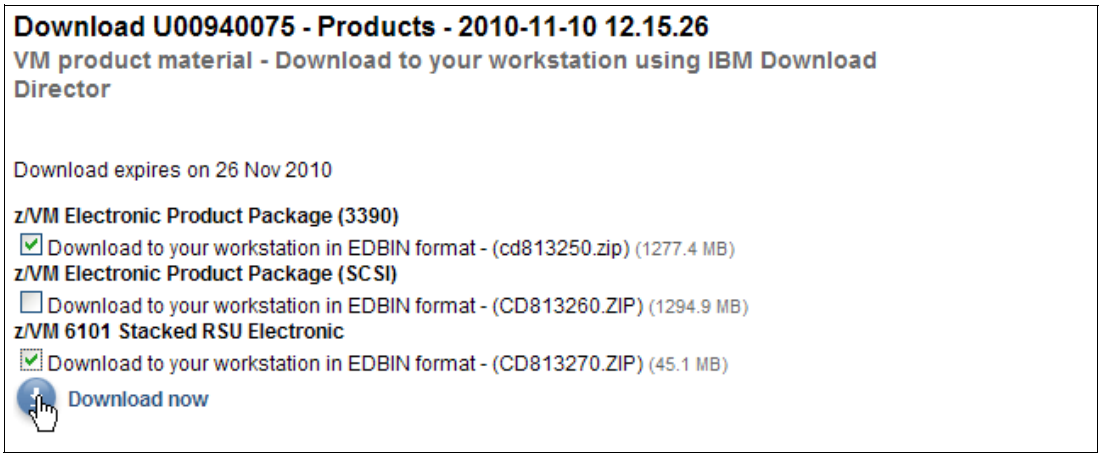


Figure 4-2 Choosing two files to be downloaded

- The z/VM install code should now be staged or ready for the FTP server to be set up. In this example where the files are staged on a Windows workstation, the two files are shown from a DOS prompt:

```
C:\zvm61> dir
...
11/11/2010  08:54 AM     1,277,435,798 cd813250.zip
11/11/2010  08:54 AM        45,088,210 CD813270.ZIP
```

- To configure an FTP server, complete all of Chapter 6 and especially 6.4, “Configuring an FTP server for z/VM installation” on page 101.

When these steps are completed, you should be able to point the z/VM installation to the FTP server that was just set up.

4.1.2 Starting the z/VM installation

This section explains how to install z/VM 6.1 from an HMC onto 3390-3 equivalent DASD. Some words are included for installing onto the larger 3390-9 DASD. For alternative configurations, such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

Perform the following steps:

- Log on to the Hardware Management Console. You should see the HMC Workplace window.
- Select the LPAR on which you want to install z/VM, often by clicking the **CPC images** icon. *Be sure* you have the correct LPAR selected. If you are not completely sure, check with someone who is.
- If necessary, click the buttons with circular arrows on the bottom right corner of the CPC Recovery menu (this is sometimes referred to as “going around the racetrack”).
- On the Recovery or CPC Recovery menu, double-click the **Integrated 3270 Console** as shown at the bottom of Figure 4-3 on page 31. A window entitled “Integrated 3270 Console for <your CPC>” will open (on older HMC levels, the window may be entitled Personal Communications).

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

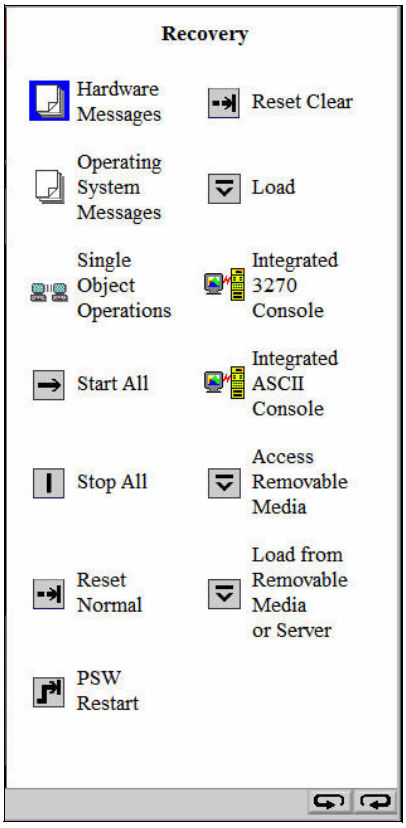


Figure 4-3 Recovery menu

- Place the z/VM Product Package Version 6 Release 1.0 DVD in the HMC DVD drive.

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

- On a z9 HMC and older, get into Single Object Operations mode by performing the following steps:
  a. Double-click the **Groups** icon in the Views Area.
  b. Double-click **Defined CPCs** in the Groups Work Area.
  c. Select your CPC.
  d. If necessary, go around the racetrack (the buttons with circular arrows on the bottom right corner) to the CPC Recovery menu.
  e. Double-click the **Single Object Operations** icon. Click **Yes** to confirm. Now a new window, Primary Support Element Workplace, should appear (on older HMC levels it will be a “window within a window”). A window about a certificate not being valid may appear. If so, click **OK**.
  f. Double-click **Groups** near the top of this window.
  g. Double-click **Images** in the Groups Work Area.

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

- The LPAR that z/VM will be installed into should still be selected. On the right you should still see the (CPC) Recovery menu. Double-click the **Load from Removable Media** or **Server** icon.

  If you received the z/VM product electronically, you will need to create your own DVDs. This step is not covered in this book. See the z/VM manual *Installation Instructions for Electronically Delivered IBM z/VM Operating System Deliverable*, GI11-2900, on the Web at:

  `http://www.vm.ibm.com/install/prodinst.html`

  If the DVD is not burned correctly, you may see the error message:

  ```
  ACT3620I "An error has occurred while trying to obtain a list of the software
  that can be loaded. ...".
  ```

  Further, this error may have the side effect of locking the DVD drive. The HMC may need to be rebooted. To prevent this from happening, be sure you create the DVDs correctly. Use newer copies of DVD-burning software that has an option for the ISO9660 format, which is recommended.

- On the Load from Removable Media or Server window shown in Figure 4-4, the **radio button Hardware Management Console CD-ROM/DVD** should be selected.
- In the same Load from Removable Media or Server window, fill in File Location with `/cpdvd`.This is the directory on the DVD with the z/VM 6.1 installation code. Click **OK**.

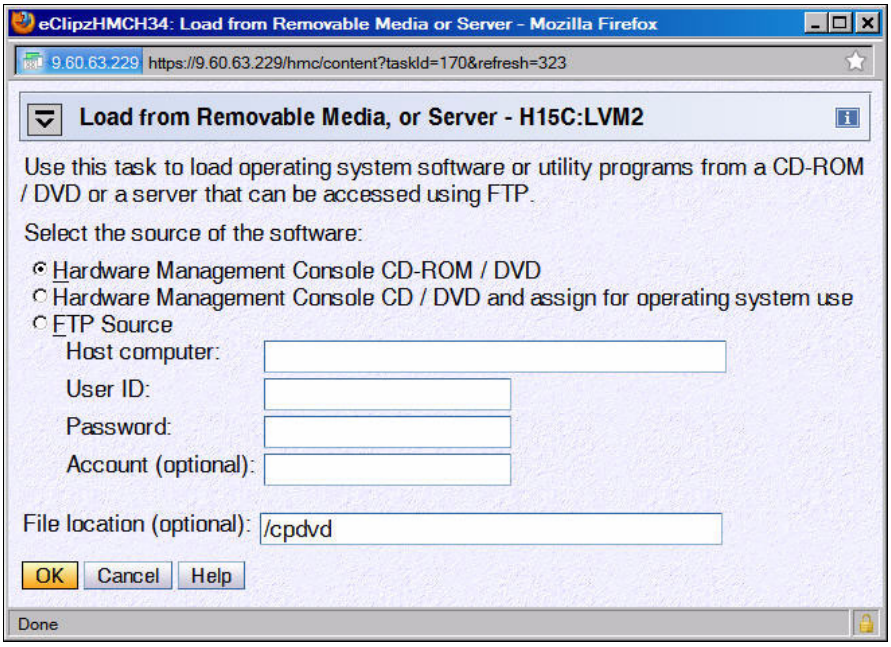


Figure 4-4 Load from Removable Media or Server panel

If you do not have physical DVDs, but there is an FTP server set up with the z/VM install code, then you can use FTP as an install method. If such an FTP server is set up, you can click **FTP Source** and fill in the fields Host Computer, User ID, Password and File location, as shown in Figure 4-5 on page 33.

Setting up an FTP server so as to provide the z/VM product files for installation is described in 6.4, “Configuring an FTP server for z/VM installation” on page 101.

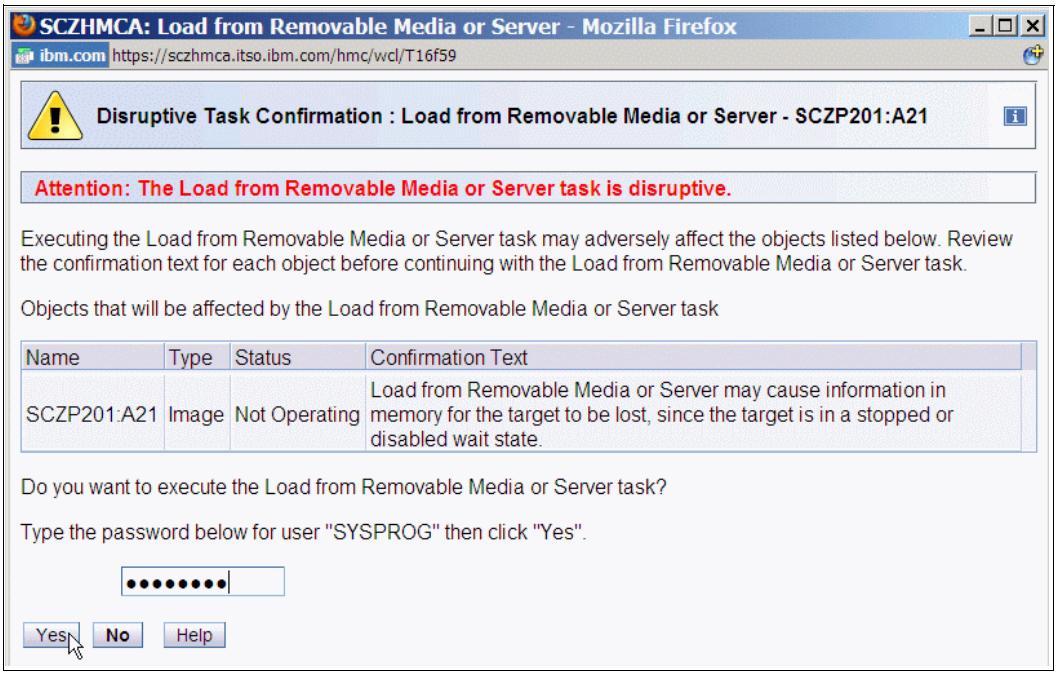


Figure 4-5 Load from Removable Media or Server panel with FTP source

- Load the RAMDISK

  a. From the Load from Removable Media or Server panel, the file `610vm.ins` should be selected as shown in Figure 4-6. Click **OK**. If you are at the HMC installing from DVD, you should see the green light on the DVD drive light up.

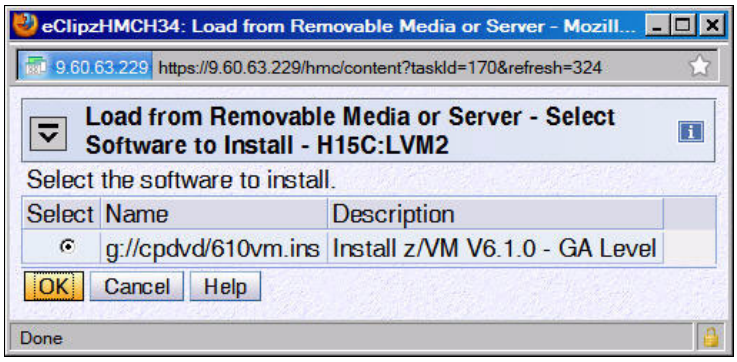


Figure 4-6 Selecting z/VM 6.1 RAMdisk system

  b. On the “Confirm the action” window, click **Yes**.

  c. You should see the Disruptive Task Confirmation: Load from CD-ROM, DVD or Server Progress window. You will be prompted for the password, as shown in Figure 4-7.

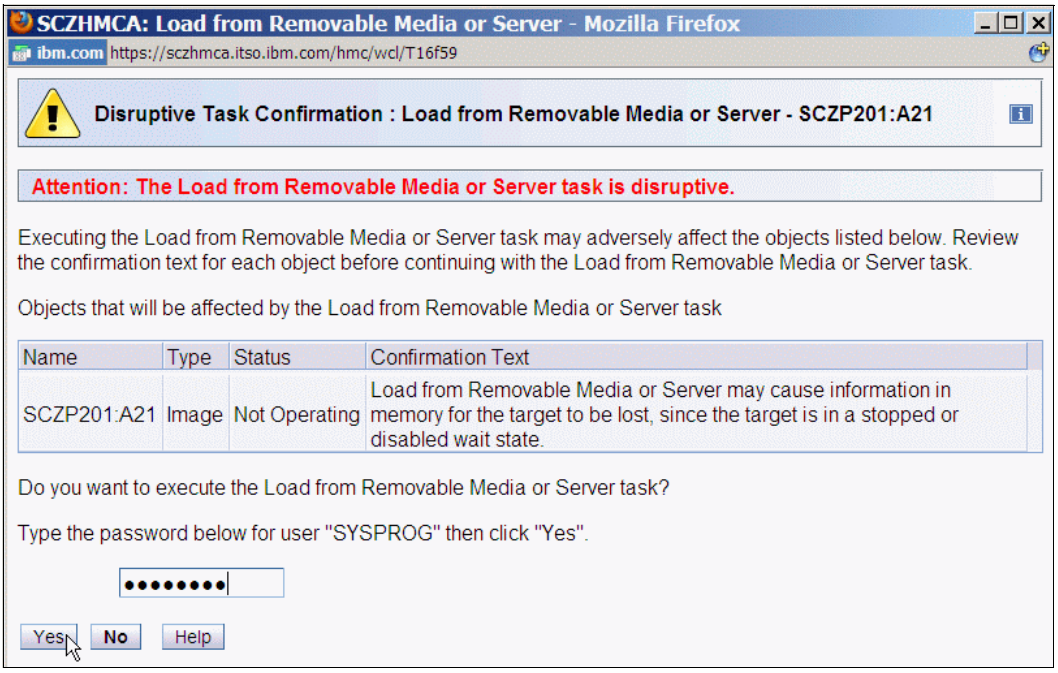


Figure 4-7 Supplying password for disruptive task

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

You should now have an in-memory z/VM 6.1 system running.

4.1.3 Copying a vanilla z/VM system to DASD

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

- You can now get out of Single Object Operations mode (if you are in it). To do so, log off the primary SE window by closing that window.
- Move to the Integrated 3270 Console window (you can use the Alt-Tab sequence). The RAMdisk should IPL and you should see z/VM boot as shown in Figure 4-8 on page 35. If the Integrated 3270 Console window is still blank, be patient, it may take a minute or two to initialize.

  Note: The Esc key in the upper left clears the Integrated 3270 console on the HMC.

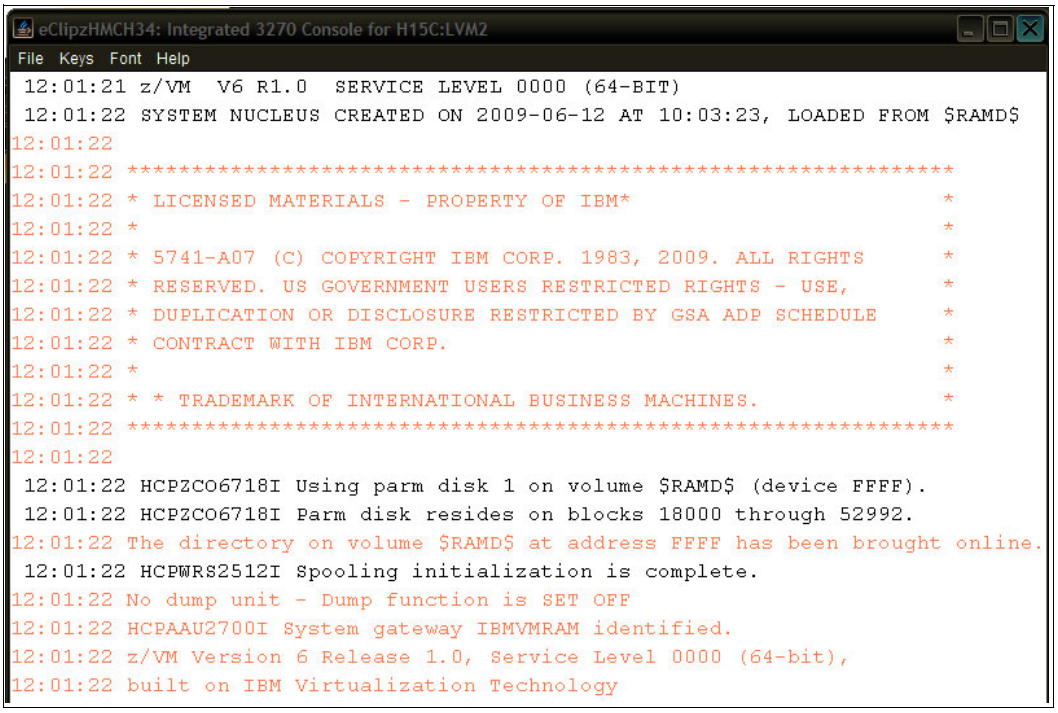


Figure 4-8 z/VM first boot on the Integrated console

- 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 (Figure 4-9):

  ```
  ==> instplan
  ```

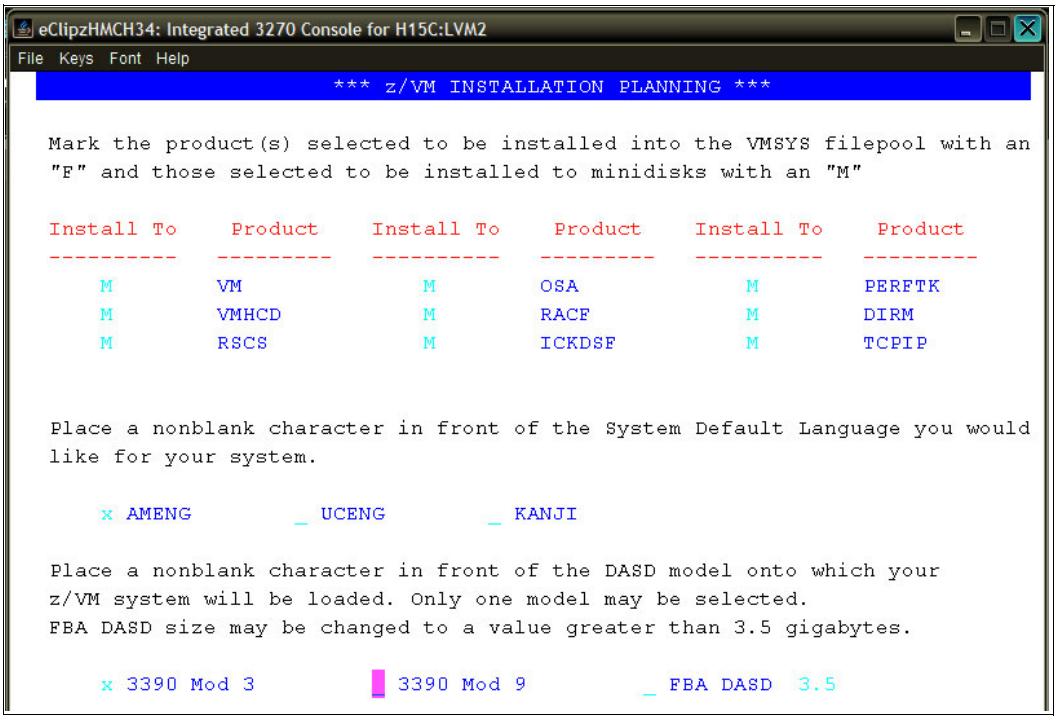


Figure 4-9 Installation planning panel

- You may need to clear the window with the Esc key. You should then see the display as shown in Figure 4-9. It is recommended that you leave the Ms in the top section alone.
- Type the letter x next to AMENG (or select your language) and 3390 Mod 3 (or the type of DASD you will use), as shown in Figure 4-9 on page 35. You can use the Tab key to move to the next input field.
- Press F5. You should see the message HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY after a list of what will be installed.
- Attach the DASD devices onto which z/VM will be installed defined in your planning worksheet in 2.7.2, “z/VM DASD worksheet” on page 17. In this example, the devices are 6280-6284.

```
==> att 6280-6284 *
6280-6284 ATTACHED TO MAINT
```

Important: The devices ***6280-6284*** are in bold italics to signify that you should replace the example value with the correct value for your site. For example, if you are installing z/VM onto DASD 1200-1204, you would type the following:

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

This convention is used throughout the book.

Running INSTDVD

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

- Execute INSTDVD:

```
==> instdvd
```

- If you are using 3390-3s, you see a panel asking for the five volumes, as shown in Figure 4-10 (if you are using 3390-9s, you will only see three lines).

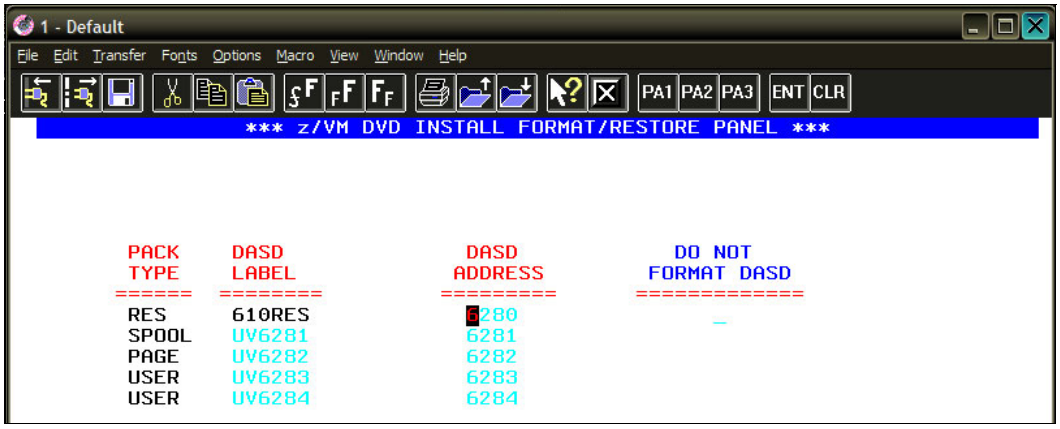


Figure 4-10 INSTDVD DASD address panel

a. Enter the addresses of the five volumes (or three for 3390-9s) that z/VM will be installed on. The labels for the last four volumes are changed because the LPAR in this example had access to other z/VM systems. Changing the labels prevents the problem described in 4.11, “Relabeling system volumes” from occurring.

b. Press F5 to start the installation.

- Verify that the five DASD addresses to be installed onto are correct. When you see the question DO YOU WANT TO CONTINUE?, type y. You should see the message NOW FORMATTING DASD ***6280***.

Important: INSTDVD can take from 45 minutes to two hours. Now may be a good time to go to Chapter 6 to set up an NFS server.

Also, read errors have been observed resulting in INSTDVD failing. If this is the case, you can try the command **`instdvd (restart`** and the install process should pick up where the read error occurred. This can be caused by dirt or fingerprints on the DVD.

- You are asked to place the system RSU in the drive. Insert the z/VM Stacked Recommended Service Upgrade 6101 DVD into the HMC DVD-ROM drive
- At the Integrated 3270 Console, type `go`. You should see a message of the form `DVDLOAD: LOADING FILE CKD5000x IMAGE *.` This step should take two to four minutes.
- Finally, you should see the message `HCPIDV8329I INSTDVD EXEC ENDED SUCCESSFULLY`.

4.1.4 IPL the vanilla z/VM from DASD

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

- In the HMC Workplace window, your LPAR should still be selected. If not, select your LPAR by clicking it. You may have to first double-click **Groups**.
- You should see the Recovery menu. Double-click the **Load** icon in the menu at the right side.
- The Load window opens, as shown in Figure 4-11 on page 38. Follow these steps:
  a. Set the load address to the new system residence (`610RES`) volume, which is ***6280*** in this example.
  b. Set the load parameter to `SYSG`. This specifies to use the Integrated 3270 console.
  c. Click **OK** to IPL.

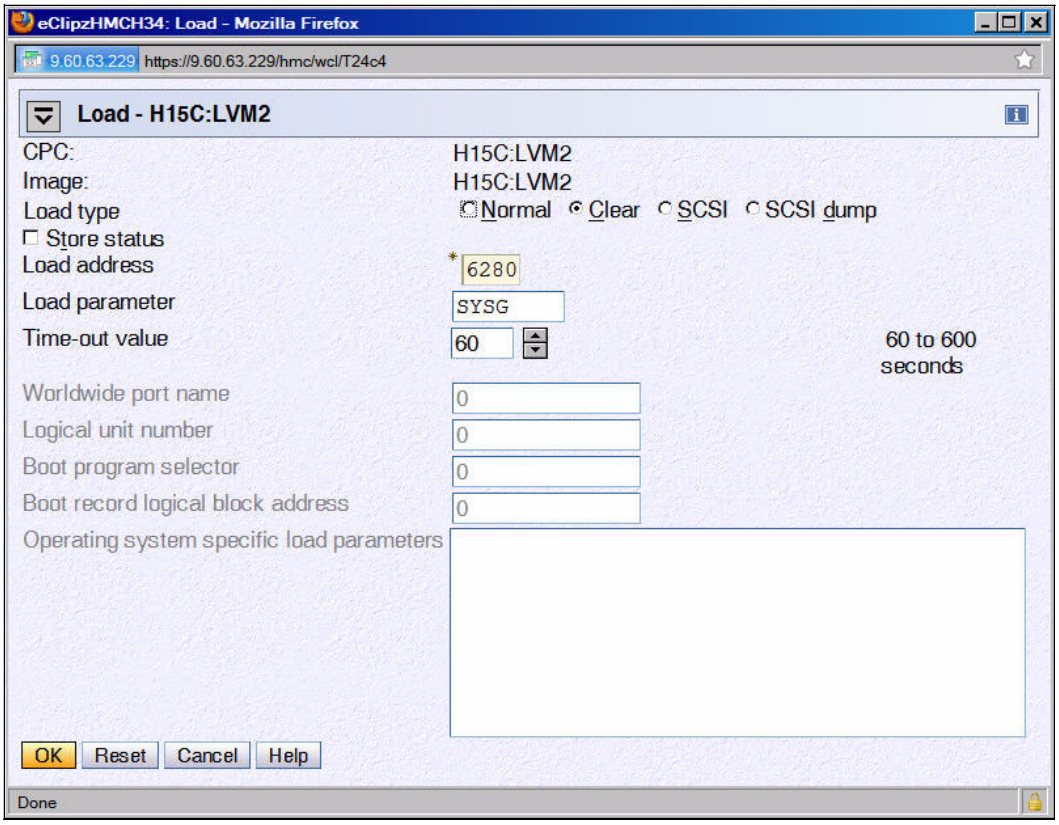


Figure 4-11 Load window

- When you see the Load Task Confirmation window, click **Yes**.
- After 1-3 minutes you should see a status of Success in the Load Progress window. Click **OK**.
- Move back to the Integrated 3270 Console window. You should see the Standalone Program Loader panel as shown in Figure 4-12 on page 39.
  a. Press the Tab key to get to the IPL Parameters section and enter the value `cons=sysg`. This specifies to use the Integrated 3270 console.

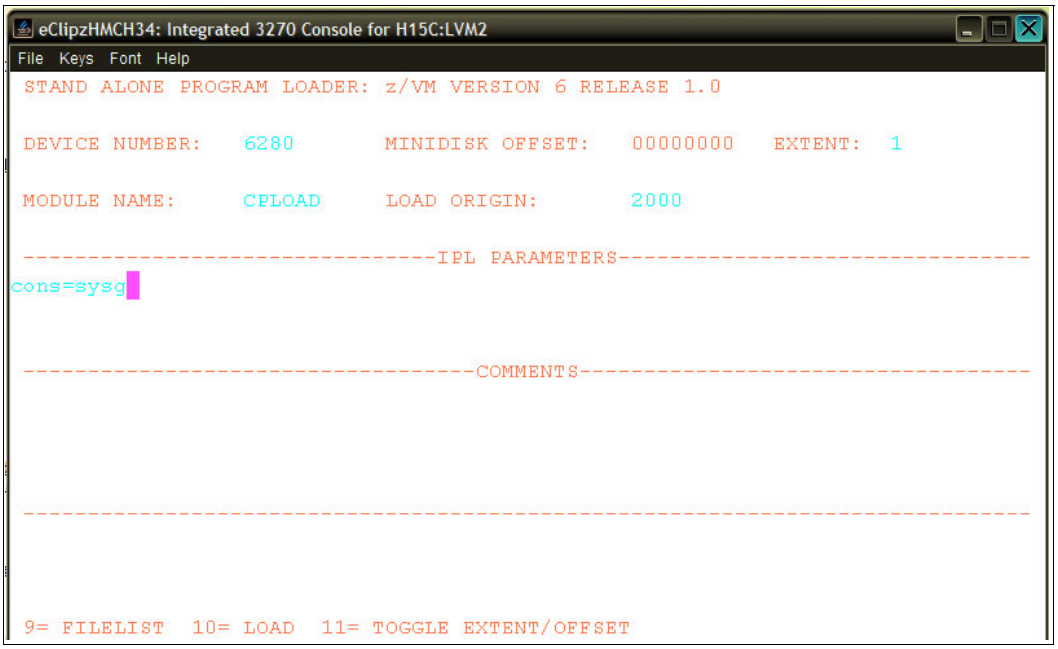


Figure 4-12 Stand Alone Program Loader

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

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

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

- At the Change TOD clock prompt, enter:

  ```
  ==> no
  ```

- The last message should be `HCPCRC8082I EREP records are accumulating for userID EREP`. Disconnect from the `OPERATOR` user ID using the **`disconnect`** command:

  ```
  ==> disc
  ```

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

4.1.5 Completing the z/VM installation

Follow these steps to complete the z/VM installation

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

  Important: When logging onto a z/VM user ID that runs CMS, you should usually press Enter at the `VM READ prompt`. Doing so will run the PROFILE EXEC and result in a prompt of the form:

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

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

  ```
  ==> ipl cms
  ```

==> Press Enter at the VM READ prompt

- Run the **instvm dvd** command:

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

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

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

```
==> ipl cms
```

  ==> Press Enter at the VM READ prompt

```
Ready;
```

- For z/VM 6.1, the service name is `6101RSU1`. Verify that this file exists on the `MAINT 500 disk`:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * * c
6101RSU1 SERVLINK C1
```

- Run the **service all** command to apply the service:

```
==> service all 6101rsu1
...
```

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

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

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

```
==> ipl cms
```

  ==> Press Enter

```
Ready;
==> put2prod
```

  This step should take about 2-4 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 on z/VM, a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

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

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

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

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

4.2 Configuring TCP/IP

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

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

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

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

4.2.1 Use the IPWIZARD tool

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

- Access the MAINT 193 disk:

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

- Invoke **IPWIZARD**.

  ```
  ==> ipwizard
  ```

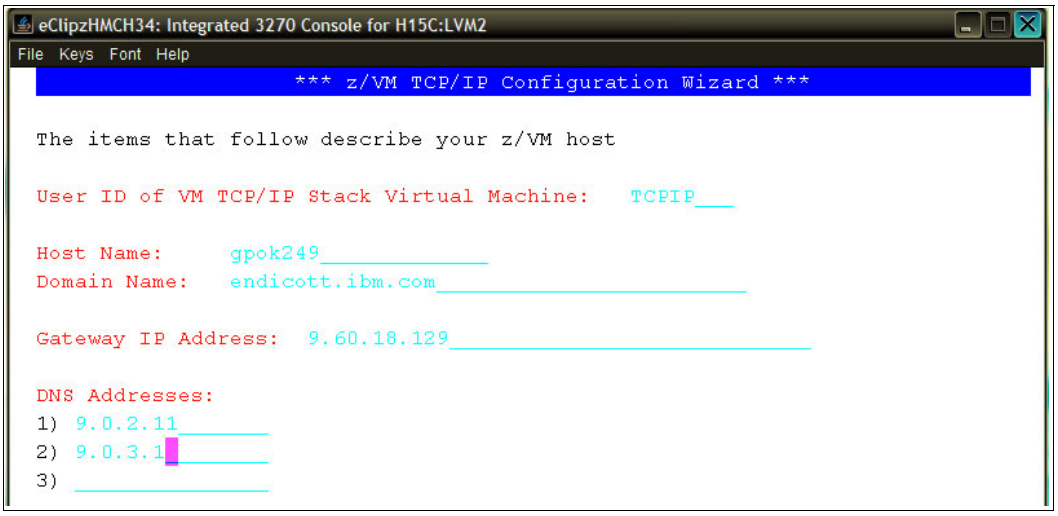


Figure 4-13 IPWIZARD panel 1

- The z/VM TCP/IP Configuration Wizard opens as shown in Figure 4-13. The first field, User ID, should always be TCPIP. Obtain the remaining values from 2.7.1, “z/VM resources worksheet” on page 16 and press F8.

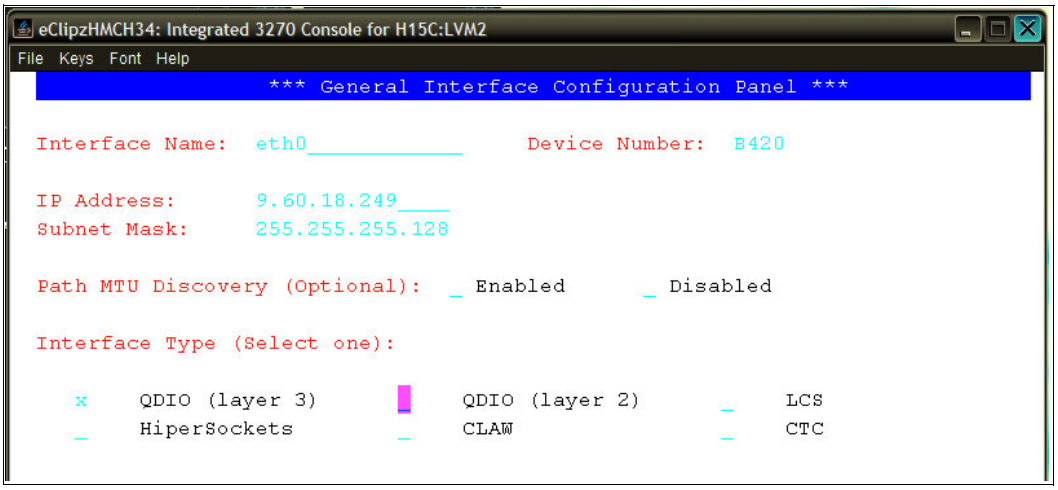


Figure 4-14 IPWIZARD panel 2

- An Interface Name of `eth0` (Figure 4-14) 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 (layer 3)` with modern OSA devices. When completed, press F8.

  Note: to utilize QDIO (layer 2), certain prerequisites must be met. Consult with the system administrator.

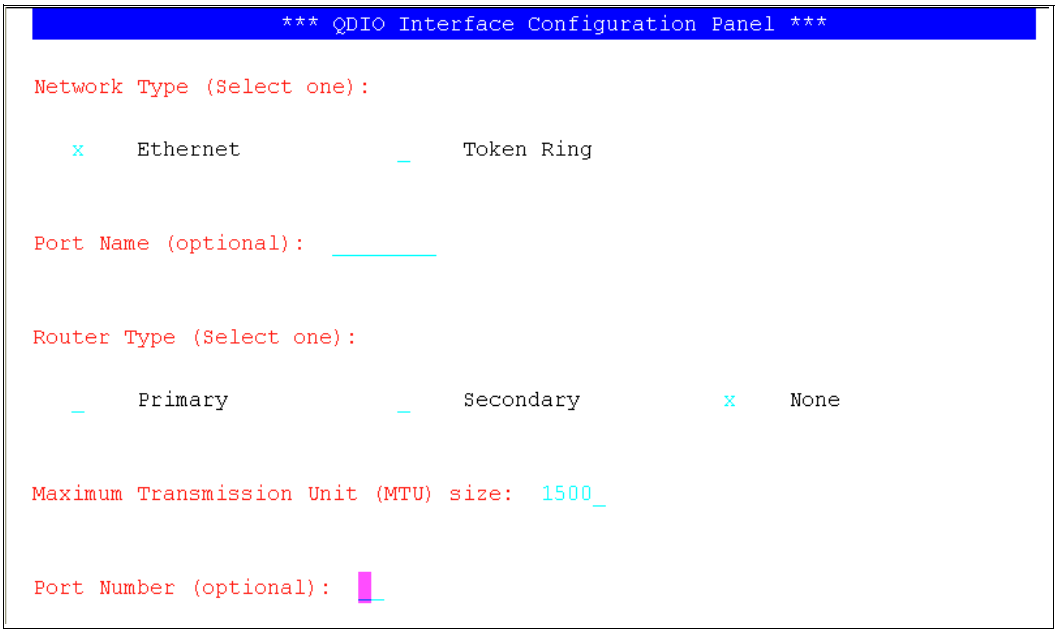


Figure 4-15 IPWIZARD panel 3

- In general, a value for the `Port Name` (Figure 4-15) is no longer necessary. Press F5 to complete the wizard.

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

- Enter `1` to restart the TCP/IP stack (you may see other warnings):

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 2     FORCED BY MAINT
...
Successfully PINGed Interface (9.12.5.22)
Successfully PINGed Gateway (9.12.4.1)
Successfully PINGed DNS (9.12.6.7)
DTCIPW2519I Configuration complete; connectivity has been verified
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCPIP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

- At this point your z/VM TCP/IP stack should be up. You should now be able to ping it from another system.

  If the IPWIZARD exec 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.

HMC Integrated 3270 Console or 3270 emulator? At this point z/VM should be accessible over the network. You can continue working at the HMC, or you can access your new system using a 3270 emulator. See 3.3, “3270 emulators” on page 24 for some brief words on that subject.

If you want to switch to the 3270 emulator, first **LOGOFF** of MAINT or **DISConnect** on the Integrated 3270 Console.

If you log off, the session is ended—it is analogous to shutting and powering down a PC. If you disconnect, your session remains where it is and is resumed when you log back on. It is analogous to turning a PC’s monitor off. In general, you should **LOGOFF** of system administration user IDs such as MAINT. However, you should always **DISConnect** from z/VM service machines such as TCPIP and user IDs running Linux. Logging off of these will terminate the service or crash Linux.

4.3 Configuring the XEDIT profile

Log on to MAINT if you are not already logged on.

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

If you have never used **XEDIT** before, there is a cheat sheet in Appendix A.1.1, “XEDIT cheat sheet” on page 246. The z/VM 6.1 PDF library is on the web at:

http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61

Search for the XEDIT User’s Guide and Command Reference. Also, there is an old manual available at:

http://ukcc.uky.edu/ukccinfo/391/xeditref.html

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

```
SET PF12 FILE
```

After:

```
SET PF12 ?
```

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

4.4 Customizing the SYSTEM CONFIG file

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

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

To make these changes, perform the following steps:

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

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  610RES 6280 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  610RES 6280 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  610RES 6280 CKD         279        398
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF2 MAINT    OCF2  B   R/O  610RES 6280 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  610RES 6280 CKD         279        398
```

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

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

- Make a backup copy of the vanilla SYSTEM CONFIG file using the **COPYFILE** command with the **OLDDATE** parameter so that the timestamp of the file is not modified. Note that because

the target file name (system) and mode (f) are the same, the equal sign (=) can be used as a wildcard.

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

- Edit the original file:

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

- The system name is set to ZVMV6R10 by default in the System_Identifier_Default statement. You can search for it using the / subcommand:

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

Modify this to the new name of your system. In this example ***POKSND61*** is used.

```
System_Identifier_Default POKSND61
```

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

```
Features ,
  Disable ,                       /* Disable the following features */
    Set_Privclass ,               /* Disallow SET PRIVCLASS command */
    Auto_Warm_IPL ,               /* Prompt at IPL always           */
    Clear_TDisk   ,               /* Don't clear TDisks at IPL time */
  Retrieve ,                      /* Retrieve options               */
    Default  99 ,                 /* Default.... default is 20      */
    Maximum  255 ,                /* Maximum.... default is 255     */
  MaxUsers noLimit ,              /* No limit on number of users    */
  Passwords_on_Cmds ,             /* What commands allow passwords? */
    Autolog  yes ,                /* ... AUTOLOG does               */
    Link     yes ,                /* ... LINK does                  */
    Logon    yes ,                /* ... and LOGON does, too        */
  Disconnect_Timeout off ,        /* Don't force disconnected users */
  Vdisk ,                         /* Allow VDISKS for Linux swaps   */
    Syslim  infinite  ,
    Userlim  infinite
```

- Define a VSWITCH:

Use the **BOTTOM** subcommand to go to the bottom of the file. Add some lines (you can use the **XEDIT** add subcommand **a3**). Define a VSWITCH and set the MAC address prefix. This sets the first three bytes of the MAC address created for each virtual NIC. If you have multiple z/VM systems, increment this value to avoid having identical MAC addresses created. The last three bytes of the MAC address are automatically incremented by z/VM as they are assigned, so they will be unique on each z/VM system. Modify the two starting addresses of the OSA triplets (***B440*** and ***B424*** in this example) to those you specified in 2.7.1, “z/VM resources worksheet” on page 16.

```
====> bot
====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev B440 B424
vmlan macprefix 020001
```

- Save your changes with the XEDIT **FILE** subcommand:

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

- 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.

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

```
==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
```

- Verify that the CP disk A has been accessed using the **QUERY CPDISK** command:

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  610RES 6280 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  610RES 6280 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  610RES 6280 CKD         279        398
```

  Note that all three CP disks are now accessed.

4.5 Configuring TCP/IP to start at IPL time

Configure the TCPIP service machine to be started when z/VM IPLs. This is commonly accomplished from AUTOLOG1's PROFILE EXEC. If the **noautolog** parameter is *not* specified when z/VM is IPLed, the AUTOLOG1 virtual machine is started. Because this virtual machine IPLs CMS, the PROFILE EXEC that is found on its A disk is run. This is analogous to the `/etc/profile` file on Linux and the `autoexec.bat` on DOS systems.

- Log off of `MAINT`:

```
==> log
```

- You should see a new logon panel. Log on to AUTOLOG1. Again the password is the same as the user ID.

- At the `VM READ` prompt enter the command **ACCESS (NOPROF** so that the `PROFILE EXEC` is not run.

```
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 09:29:16 EST FRIDAY 11/20/09
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V6.1.0    2009-11-19 13:47

==> acc (noprof
```

- Copy the PROFILE XEDIT from the `MAINT 191` disk so that 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
```

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

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

- You should see the text in the top half of the following example. Modify it as follows.

a. You can safely delete the `Address Command` line.

b. Add a line to start the `TCPIP` user ID using the **`XAUTOLOG`** command and keep two statements that start the VSWITCH cloners.

c. Add a line to log off of `AUTOLOG1` when the EXEC is complete. There is no need to keep that virtual machine running because its sole purpose is to run the PROFILE EXEC.

Before:

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

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

After:

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

- Save your changes with the **`FILE`** subcommand

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

- Log off of `AUTOLOG1`:

```
==> log
```

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

4.5.1 Renaming the TCPIP configuration file

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

- Log on to TCPMAINT. The PROFILE TCPIP file is on the TCPMAINT 198 disk, which is accessed as the D disk.
- Make a backup copy of the original PROFILE TCPIP, then rename it to <SYSTEM_ID> TCPIP (where <SYSTEM_ID> is ***POKSND61*** 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 poksnd61 = =
```

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

```
==> listfile * * d
POKSND61 TCPIP    D1
PROFILE  $TCPBAK  D1
SYSTEM   $DTCBAK  D1
SYSTEM   DTCPARMS D1
TCPIORIG PROFILE  D1
```

4.5.2 Copy the PROFILE XEDIT file

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

- 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
```

- Copy the PROFILE XEDIT file to your A disk:

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

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

4.5.3 Configuring the FTP server

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

- Edit the file

```
==> x poksnd61 tcpip d
```

- 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:

```
==> x poksnd61 tcpip d
```

Before:

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

After:

```
; ------------------------------------------------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE ROUTED DHCPD REXECD SNMPD SNMPQE
ENDOBEY
; ------------------------------------------------------------------
AUTOLOG
  FTPSERVE  0
ENDAUTOLOG
PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
; 25   TCP SMTP                ; SMTP Server
...
```

- Save your changes with the **FILE** subcommand:

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

You could continue to configure the system, but at this time it is recommended that you test your changes by shutting down and re-IPLing the system.

4.5.4 Shutting down and re-IPLing the system

You may want to be able to shut down and re-IPL 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 re-IPL 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 re-IPL 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.

To shut down and re-IPL the system, perform the following steps:

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

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

  You will lose your session, but it should come back in a few minutes as described above.

- When your system is back, start a 3270 session and log on as MAINT. This shows that there is TCP/IP access to z/VM.

Important: If you cannot start another 3270 session, do not despair, and consider this a good learning experience. You must go back to an Integrated 3270 session from the HMC. Verify that TCPIP is logged on. If it is logged on and you still cannot get to your system, log TCPIP off (or just re-IPL CMS), log back on, press Enter and watch the messages for errors.

- Query the new VSWITCH:

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-01-00-00-01
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
  RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
  RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1  BACKUP
```

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

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

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

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

4.6 Adding paging volumes

The z/VM operating system resides on the first three CP volumes (or one volume if installing onto 3390-9s). z/VM 6.1 is installed 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 is probably not.

It is recommended that you add at least three paging volumes, giving you a total of four (or one more 3390-9). Having adequate paging space will give you plenty of *headroom* to add more Linux virtual machines. A rule of thumb for the amount of paging space is to have twice as much as the total of all memory for all running Linux user IDs combined.

4.6.1 Formatting the paging volumes

Before adding paging volumes to the system, the DASD volumes to be used for minidisk space (PERM) and paging space (PAGE) must be formatted. Normally this is done one volume at a time using 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 B.1.1, "The CPFORMAT EXEC" on page 250. 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, or you can modify the REXX EXEC.

Getting the CPFORMAT EXEC to z/VM

Perform the following steps:

- Log off of MAINT so you will be able to get the MAINT 191 disk in read-write mode using FTP.

  Important: At this point, you need access to the NFS server described in Chapter 6, in order to get the files CPFORMAT EXEC. If you did not complete that chapter, it is necessary to do so in order to proceed.

- Start an SSH (putty) session to the NFS server and change to the vm/ directory, which was created when you untarred the files associated with this book. Verify that the file CPFORMAT.EXEC exists:

```
# cd /nfs/virt-cookbook-RH6/vm
# ls cpformat*
cpformat.exec
```

- Start an FTP session to z/VM. If you get a reply from the FTP server, it shows that you correctly configured it on the z/VM TCPMAINT user ID. Issue the **PUT** subcommand to copy the file.

```
# ftp 9.60.18.249
Name (9.12.5.22:root): maint
331-Password: maint
230-MAINT logged in; working directory = MAINT 191
...
ftp> put cpformat.exec
...
ftp> quit
```

You should now have the CPFORMAT EXEC on the MAINT 191 disk.

Using the CPFORMAT EXEC

To use the CPFORMAT EXEC, perform the following steps:

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

```
==> cpformat ?

Synopsis:

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

```
    <xxxx> is the 4 digit address

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

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

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

- The DASD that will be used for paging volumes in this example are at real addresses ***6285***, ***6286***, and ***6287***. Query the DASD devices to see their status:

```
==> q 6285 6286 6287
DASD 6285 UM6285  , DASD 6286 UM6286  , DASD 6287 UM6287
```

- Attach the devices to MAINT (the last parameter of * means the current user ID) using the **ATTACH** command:

```
==> att 6285-6287 *
6285-6287 ATTACHED TO MAINT
```

- Use the **CPFORMAT** command with the **AS PAGE** parameter:

```
==> cpformat 6285-6287 as page

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6285 MAINT    6285 3390  FR6285 6285          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6286 MAINT    6286 3390  FR6286 6286          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6287 MAINT    6287 3390  FR6287 6287          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    6285 MAINT    6285 3390  UP6285 6285          0       3339
MAINT    6286 MAINT    6286 3390  UP6286 6286          0       3339
MAINT    6287 MAINT    6287 3390  UP6287 6287          0       3339
```

This formatting job should run for about 10-30 minutes, depending on many factors.

4.6.2 Formatting DASD for minidisks

In addition to CP disks such as page space, system disks are needed to create minidisks for the virtual machines. In this section the DASD that will be used for the minidisks of LNXMAINT, RH6CLONE and RH6GOLD will be formatted.

- Query the DASD that will be used for minidisks. In this example they are ***6289***, ***6290***, ***6293***, ***6294*** (3390-3s), ***63A2*** and ***63A9*** (3390-9s):

```
==> q 6289 6290 6293 6294 63a2 63a9
DASD 6289 FR6289  , DASD 6290 FR6290  , DASD 6293 FR6293  , DASD 6294 FR6294
DASD 63A2 FR63A2  , DASD 63A9 FR63A9
```

- Attach the six volumes that will be used for the cloner, the common CMS disk and the golden image. Note that in this example the DASD are four 3390-3s and two 3390-9s. If you are using all 3390-3s, you will need eight devices:

```
==> att 6289 6290 6293 6294 63a2 63a9 *
6289 6290 6293 6294 63A2 63A9 ATTACHED TO MAINT
```

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

```
==> cpformat 6289 6290 6293 6294 63a2 63a9 as perm

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6289 MAINT    6289 3390  FR6289 6289          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6290 MAINT    6290 3390  FR6290 6290          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6293 MAINT    6293 3390  FR6293 6293          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6294 MAINT    6294 3390  FR6294 6294          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    63A2 MAINT    63A2 3390  FR63A2 63A2          0      10017
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    63A9 MAINT    63A9 3390  FR63A2 63A9          0      10017

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PERM space (y/n)? y

...
DASD successfully formatted:  UM6289 UM6290 UM6293 UM6294 UM63A2 UM63A9
6289 6290 6293 6294 63A2 63A9 DETACHED
6289 6290 6293 6294 63A2 63A9 ATTACHED TO MAINT

DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    6289 MAINT    6289 3390  UM6289 6289          0       3339
MAINT    6290 MAINT    6290 3390  UM6290 6290          0       3339
MAINT    6293 MAINT    6293 3390  UM6293 6293          0       3339
MAINT    6294 MAINT    6294 3390  UM6294 6294          0       3339
MAINT    63A2 MAINT    63A2 3390  UM63A2 63A2          0      10017
MAINT    63A9 MAINT    63A9 3390  UM63A9 63A9          0      10017
```

You should now have newly formatted volumes that can be used for minidisks.

4.6.3 Updating the SYSTEM CONFIG file

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

- Log on to MAINT.

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

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  610RES 61A2 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  610RES 61A2 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  610RES 61A2 CKD         279        398
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
```

  It is good to remember this sequence of steps.

- Make a copy of the working SYSTEM CONFIG file using the “WRKS” (it works!) suffix convention:

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

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

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

    CP_Owned   Slot   1  610RES
    CP_Owned   Slot   2  UV6281
    CP_Owned   Slot   3  UV6282
    CP_Owned   Slot   4  UV6283
    CP_Owned   Slot   5  UV6284
    CP_Owned   Slot   6  UP6285
    CP_Owned   Slot   7  UP6286
    CP_Owned   Slot   8  UP6287
    CP_Owned   Slot   9  RESERVED
    CP_Owned   Slot   10  RESERVED
    CP_Owned   Slot   11  RESERVED
...
```

- Move down to the User_Volume_List section. User volumes (PERM) can be specified individually with the User_Volume_List statement, or with wild cards using the User_Volume_Include statement. If you are using the labelling convention enforced by the CPFORMAT EXEC and no other LPAR will be using the same volumes with the same prefix, then add the following single line to include all PERM space as volume labels all beginning with UM6.

```
====> /user_v
/*************************************************************************/
/*                          User_Volume_List                             */
/* These statements are not active at the present time.  They are        */
/* examples, and can be activated by removing the comment delimeters     */
/*************************************************************************/
User_Volume_Include UM6*
/*  User_Volume_List USRP01    */
/*  User_Volume_List USRP02    */
```

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

Important: If other z/VM LPARs might be attaching volumes with the UM prefix, you should specifically list each volume to be attached to `SYSTEM` using the `User_Volume_List statement`. This will prevent the possibility of multiple z/VM systems writing to the same volume. In this example, the list would be:

```
User_Volume_List UM6289
User_Volume_List UM6290
User_Volume_List UM6293
User_Volume_List UM6294
User_Volume_List UM63A2
```

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

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

- When you have confirmed that there are no syntax errors, put the MAINT CF1 disk back online. The following example shows how you did this previously:

```
==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  610RES 0200 CKD          39         83
MNTCF2 MAINT    OCF2  B   R/O  610RES 0200 CKD          84        128
MNTCF3 MAINT    OCF3  C   R/O  610RES 0200 CKD         129        188
```

4.6.4 Testing the changes

It is recommended that you again shut down and re-IPL to test the changes. Before you shut down, note that you have only one page volume (`UV6282` in this example) using the **QUERY ALLOC PAGE** command. Your output should look similar to the following:

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
UV6282 6282          1       3338 600840      1      4   1%
                                  ------ ------        ----
SUMMARY                           600840      1          1%
USABLE                            600840      1          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 using 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, log on as MAINT and look at the page space again. You should now see that you have six paging volumes:

```
==> q alloc page
                EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
UV6282 6282          1       3338 600840      1      5   1%
UP6285 6285          0       3338 601020      0      0   0%
UP6286 6286          0       3338 601020      0      0   0%
UP6287 6287          0       3338 601020      0      0   0%
                                  ------ ------        ----
SUMMARY                            2348K      1          1%
USABLE                             2348K      1          1%
```

The output shows that there are four paging volumes constituting 2348 K pages, or about 9 GB of page space (a page is 4 KB).

4.7 Creating a user ID for common files

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

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

4.7.1 Define the user in the USER DIRECT file

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

- Edit the USER DIRECT file and add the following user ID definition to the bottom of the file. A comment is added signifying the split between z/VM system user IDs and locally added user IDs (this can be helpful when moving to a new version of z/VM):

```
==> x user direct c
====> bottom
====> a 9
...
*-------------------------------------------------------------
* z/VM system user IDs are above, local user IDs are below
*-------------------------------------------------------------
USER LNXMAINT LNXMAINT 64M 128M BEG        1
 INCLUDE TCPCMSU                           2
 LINK TCPMAINT 592 592 RR                  3
 MDISK 0191 3390 0001 0020 UM6289 MR READ     WRITE    MULTIPLE   4
 MDISK 0192 3390 0021 0300 UM6289 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 64 MB, with class B, E, and G privileges.

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

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

4 Define a 191 minidisk of size 20 cylinders from volume UM6289.

5 Define a 192 minidisk of size 300 cylinders (approximately 225 MB) from volume UM6289 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.

- Whenever an MDISK statement is added or modified in the USER DIRECT file, always check for overlapping cylinders and gaps (gaps will only leave empty disk space; however, overlaps can occur because 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.
```

- 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
```

- 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 all the gaps using the **ALL** subcommand. You should see some gaps:

```
====> all /gap
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  216  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  86  line(s) not displayed --------------------
```

Type all with no argument again to get out of this mode

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

Three gaps should be listed on the right side:

- 501 cylinders on the $$$$$$ volume
- 1 cylinder on the $$$LNX volume
- 1 cylinder on volume used for LNXMAINT 191 and 192 disks (UM6289 in this example)

You do not have to worry about the first two gaps because they are expected, given the layout of the default USER DIRECT file. To avoid a 1-cylinder gap being reported on each user volume, we recommend 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.

- Get out of the file USER DISKMAP with the **QUIT** command or by pressing F3.
- Edit the USER DIRECT file again and add a new minidisk definition at virtual address A04 for the first cylinder of the DASD you added (the label is UM6289 in this example):

```
==> x user direct
====> /user $alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 610RES R
 MDISK A02 3390 000 001 UV6283 R
 MDISK A03 3390 000 001 UV6284 R
 MDISK A04 3390 000 001 UM6289 R
```

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

```
==> diskmap user
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.
==> x user diskmap
====> prefix off
====> all /gap
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  303  line(s) not displayed --------------------
```

- When you are done you can quit by pressing F3.

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

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
```

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

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

4.7.2 Logging and customizing the new user ID

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

- Log off of MAINT and log on to LNXMAINT.

```
LOGON LNXMAINT
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
```

```
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:14:38 EST FRIDAY 11/20/09
z/VM V6.1.0    2009-11-19 13:47

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.

- 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 the following example:

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

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

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

- You have now formatted the two minidisks and accessed them as file modes `A` and `D`. You can confirm this with the **`QUERY DISK`** command:

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
LNX191 191  A   R/W    20 3390 4096        0          7-00       3593
3600
LXM192 192  D   R/W   300 3390 4096        0         11-00      53989
54000
MNT190 190  S   R/O   100 3390 4096      694      15028-83       2972
18000
MNT19E 19E  Y/S R/O   250 3390 4096     1021      28254-63      16746
45000
```

4.7.3 Copying a PROFILE XEDIT

Copy the `PROFILE XEDIT` from the MAINT 191 disk so that XEDIT sessions will have a common interface among user IDs. Perform the following steps:

- 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
```

- Copy the `PROFILE XEDIT` file to the A disk:

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

- Also copy the same file to the D disk (which will become the Linux user ID's read-only A disk). Then release and detach the MAINT 191 disk:

```
==> copy profile xedit z = = d
==> rel z (det
DASD 0120 DETACHED
```

4.7.4 Creating a PROFILE EXEC

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

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

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

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

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

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

- 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`. Verify that these three disks are accessed with the **QUERY DISK** command:

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
LXM191 191  A   R/W    20 3390 4096        2          9-01       3591
3600
LXM192 192  D   R/W   300 3390 4096        0         11-00      53989
54000
TCM592 592  E   R/O    70 3390 4096      903      10183-81       2417
12600
MNT190 190  S   R/O   100 3390 4096      694      15028-83       2972
18000
MNT19E 19E  Y/S R/O   250 3390 4096     1021      28254-63      16746
45000
```

- Verify that your F11 and F12 keys are set to the **RETRIEVE** command using the **QUERY PFKEYS** command:

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

```
PF12 RETRIEVE BACKWARD
...
```

4.7.5 Copying files associated with this book to LNXMAINT

The z/VM files associated with this book are in the vm/ subdirectory of the NFS server you set up earlier. These files should be stored on the larger 192 disk, which is accessed as your D disk. Perform the following steps:

- Log off of LNXMAINT so that the 192 disk can be accessed read-write.
- Start an SSH session on the NFS server and change the directory to the VM files associated with this book. The directory name is one of the following two, depending on the distribution you are working with:

  ```
  # cd /nfs/virt-cookbook-S11S1/vm
  ```

- FTP to z/VM. By default FTP copies files to your 191 disk, so first change directory to the LNXMAINT 192 disk. The files are all in ASCII and the default behavior is to convert to ASCII to EBCDIC. Use the **mput *** subcommand to copy the files from the vm/ directory to LNXMAINT:

  ```
  # ftp 9.60.18.249
  Connected to 9.12.5.22.
  Name (9.12.5.22:root): lnxmaint
  331-Password:
  Password: lnxmaint
  230-LNXMAINT logged in; working directory = LNXMAINT 191
  Remote system type is z/VM.
  ftp> cd lnxmaint.192
  250 Working directory is LNXMAINT 192
  ftp> prompt
  Interactive mode off
  ftp> mput *
  ...
  ftp> quit
  ```

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

  ```
  ==> filel * * d
  LNXMAINT FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
  Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
        CHPW610  XEDIT    D1 V         72        190          3  8/31/10  8:20:03
        CPFORMAT EXEC     D1 V         79        252          3  8/31/10  8:20:03
        PROFILE  EXEC     D1 V         63         17          1  8/31/10  8:20:03
        SAMPLE   PARM-S11 D1 V         80         11          1  8/31/10  8:20:03
        SLES11S1 EXEC     D1 V         68         10          1  8/31/10  8:20:03
        SWAPGEN  EXEC     D1 V         72        467          6  8/31/10  8:20:03
        PROFILE  XEDIT    D1 V         45         17          1  8/31/10  8:20:03
  ```

4.8 Customizing system startup and shutdown

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

4.8.1 Configuring the AUTOLOG1 PROFILE EXEC

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

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

To accomplish this, perform the following steps:

- Log off of LNXMAINT and log on to AUTOLOG1. At the VM READ prompt you have usually been pressing Enter, which causes the PROFILE EXEC to be run. If you do not want this EXEC to run, enter the command **ACCESS (NOPROF**:

```
LOGON AUTOLOG1
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 09:29:16 EST FRIDAY 11/20/09
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V6.1.0    2009-11-19 13:47

==> acc (noprof
```

- Make a copy of the working PROFILE EXEC:

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

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

```
==> x profile exec
/*************************/
/*  Autolog1 Profile Exec  */
/*************************/
'cp xautolog tcpip'                 /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'        /* Allow guests 5 min to shut down */
'cp logoff'                         /* logoff when done */
```

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

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

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

4.8.2 Testing the changes

To test your changes you must re-IPL z/VM again. Be sure you are in a position to do so!

Perform the following steps:

- Shut down and re-IPL your system.

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

- When your system comes back, log on as MAINT.
- Query the SRM values to see that the new STORBUF settings are in effect and the SIGNAL SHUTDOWN value is set to 300 seconds:

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

This output shows that your changes have taken effect.

4.9 Addressing z/VM security issues

This section briefly discusses the following security issues.

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

VM security products

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

High-level z/VM security

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

http://www.vm.ibm.com/library/zvmsecint.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* addresses this issue. It is on the Web at:

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

z/VM user ID and minidisk passwords

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

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

User IDs	The password required to log on 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 `CHPW610 XEDIT` macro described in the next section.

4.9.1 Changing passwords in USER DIRECT

Changing the passwords can be done manually in XEDIT. However, this is both tedious and error-prone. So an XEDIT macro named `CHPW610 XEDIT` has been included with this book. The source code is in Appendix B.1.2, “The CHPW610 XEDIT macro” on page 254.

This macro changes 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.

- Log on to `MAINT`.
- Link and access the LNXMAINT 192 disk to pick up the CHPW610 XEDIT macro:

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

- 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.

- Edit the `USER DIRECT` file with the parameter **`(profile chpw610)`** followed by the new password. Rather than invoking the default profile of PROFILE XEDIT, this command invokes the XEDIT macro named `CHPW610 XEDIT` and passes it the new password. For example, to change all passwords to `lnx4vm`, enter the following command:

```
==> x user direct c (profile chpw610) lnx4vm

Changing all passwords to: LNX4VM
```

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

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

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

- Bring the changes online with the **DIRECTXA** command:

  ```
  ==> directxa user
  z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
  EOJ DIRECTORY UPDATED AND ON LINE
  HCPDIR494I User directory occupies 43 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 because you can use the **XEDIT CHANGE** subcommand. For example, to change all passwords from `lnx4vm` to `vm4lnx`, invoke the following commands:

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

Congratulations, your z/VM system is now customized and ready for Linux.

4.10 Backing up your z/VM system to tape

Your system is now customized with a running TCP/IP stack, a highly available VSWITCH, a startup and shutdown process, and 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, `610RES`, `610SPL`, `610PAG`, `610W01`, and `610W02` (or just the first three if you are using 3390-9s). If you changed the labels of the last four at install time, then use those labels. You have also configured a sixth volume that is important to Linux: that is, the first 320 cylinders of the volume with `LNXMAINT` on it.

To back up 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.

4.11 Relabeling system volumes

In previous books, the z/VM installation was described using “standard labels” on the CP-owned volumes (for example, `610RES`, `610SPL`, `610PAG`, `610W01`, and `610W02`). In this book, changing the last four labels to include the real device address in the last four characters of each label is recommended (the label of the “res pack”, for example `610RES`, cannot be modified at install time). This alleviates the possibility that another vanilla z/VM system with the same labels is installed onto volumes accessible by your z/VM system. If that happens, it is likely that one of the systems will not IPL correctly.

To understand this possibility, refer to Figure 4-16. 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 540RES volume will be used, but the remaining volumes in the system are searched for by volume label, not by device address. Because z/VM system 1's addresses are lower than z/VM system 2's, system 2 will be using system 1's volumes. This is not good for either system!

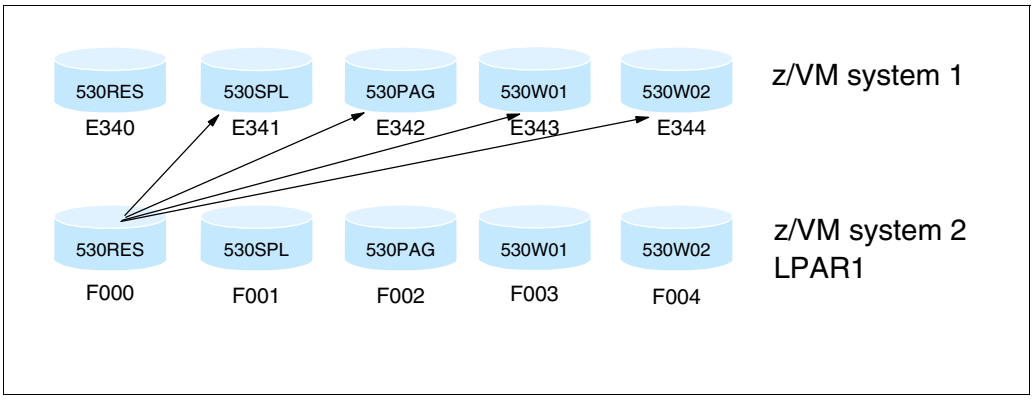


Figure 4-16 The problem with two z/VM systems with identical volume labels

In previous books a REXX EXEC and an XEDIT macro were provided to help in the process of relabeling system volumes. However, if you followed the previous steps, you will have only one standard label, 610RES. The EXEC and macro are no longer provided because they relied on standard labels. However, high-level steps are still included. If you modified all labels except for the first one at install time, it is usually not necessary to perform the steps in this section.

If you do need to relabel the system volumes, perform the following steps:

- "Modifying labels in the SYSTEM CONFIG file" on page 66
- "Modifying labels in the USER DIRECT file" on page 68
- "Changing the labels on the five volumes" on page 68
- "Shutting down your system and restarting it" on page 69

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. (Close your door, do not answer your phone or e-mail, turn off instant messaging :).)

4.11.1 Modifying labels in the SYSTEM CONFIG file

An HMC Integrated 3270 Console session will be needed in this section because z/VM will have to be restarted with a FORCE option.

- Start a 3270 session. It can be a 3270 emulator session for now, or all of the steps can be done from the HMC.
- Note the first five CP-owned volumes using the **QUERY CPOWNED** command. In this example they are D850-D854:

```
==> q cpowned
   1  610RES  D850  Own    Online and attached
   2  610SPL  D851  Own    Online and attached
   3  610PAG  D852  Own    Online and attached
```

```
  4  610W01  D853  Own    Online and attached
  5  610W02  D854  Own    Online and attached
  6  MPD855  D855  Own    Online and attached
 ...
```

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

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

- Edit the SYSTEM CONFIG file and modify the five labels (if you installed onto 3390-9s, there are only three labels, no W01 and W02 volumes are required):

```
==> x system config f
====> c/610RES/MVD850/*
DMSXCG517I 3 occurrence(s) changed on 3 line(s)
====> top
====> c/610SPL/MVD851/*
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
====> top
====> c/610PAG/MVD852/*
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
====> top
====> c/610W01/MVD853/*
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
====> top
====> c/610W02/MVD854/*
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

- Search for the string cp_owned and you should see the new labels. Be sure they are correct before saving the file with the **FILE** subcommand:

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

 CP_Owned   Slot   1  MVD850
 CP_Owned   Slot   2  MVD851
 CP_Owned   Slot   3  MVD852
 CP_Owned   Slot   4  MVD853
 CP_Owned   Slot   5  MVD854
 CP_Owned   Slot   6  MPD855
 ...
====> file
```

- Verify that 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 with the **QUERY CPDISK** command:

```
==> rel f (det
```

```
DASD OCF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
Ready; T=0.01/0.01 09:19:57
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  610RES D850 CKD          39        158
MNTCF2 MAINT    OCF2  B   R/O  610RES D850 CKD         159        278
MNTCF3 MAINT    OCF3  C   R/O  610RES D850 CKD         279        398
```

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 IPL cleanly.

4.11.2 Modifying labels in the USER DIRECT file

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

- Modify the labels in the `USER DIRECT` file. If you installed z/VM onto 3390-9s, you will need only the first three **`CHANGE`** subcommands:

```
==> copy user direct c = direwrks = (oldd rep
==> x user direct c
====> c/610RES/MVD850/*
DMSXCG517I 94 occurrence(s) changed on 94 line(s)
====> top
====> c/610SPL/MVD851/*
DMSXCG517I 78 occurrence(s) changed on 78 line(s)
====> top
====> c/610PAG/MVD852/*
DMSXCG517I 117 occurrence(s) changed on 117 line(s)
====> top
====> c/610W01/MVD853/*
DMSXCG517I 2 occurrence(s) changed on 2 line(s)
====> top
====> c/610W02/MVD854/*
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

  Traverse the file to view the changes before saving the changes with the **`FILE`** subcommand:

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

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

4.11.3 Changing the labels on the five volumes

In this section you will change the labels on the five volumes using the **`CPFMTXA`** command. Four of the five system disks are defined as full-pack minidisks to MAINT as virtual devices `122-124` (`610RES`, `610SPL`, `610W01`, and `610W02`). If you installed z/VM onto 3390-9s, you will not need to use `124` and `125`. The fifth volume, `610PAG`, is defined as the virtual device `$PAGE$ A03`. To modify the system volumes' labels, you will use these virtual addresses.

For reference, following are the entries in the USER DIRECT file:

```
...
USER $PAGE$   NOLOG
 MDISK A03 3390 000 END 610PAG R
..
MDISK 122 3390 000 END 610SPL MR
MDISK 123 3390 000 END 610RES MR
MDISK 124 3390 000 END 610W01 MR
MDISK 125 3390 000 END 610W02 MR
...
```

Perform the following steps:

- Use the **CPFMTXA** command to relabel the five system volumes (you will only need the first three if you installed onto 3390-9s). Be sure to watch for a return code of 0 on each command:

```
==> cpfmtxa 123 mvd850 label
...
==> cpfmtxa 122 mvd851 label
...
==> link $page$ a03 a03 mr
==> cpfmtxa a03 mvd852 label
...
==> cpfmtxa 124 mvd853 label
...
==> cpfmtxa 125 mvd854 label
...
```

- Now that the five volumes have been relabeled (sometimes called *clipping the volumes,* derived from a contraction of the z/OS term ***c**hange **l**abel **p**rogram*), you can run the **DIRECTXA** command to update the directory:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
Ready(00005); T=0.01/0.01 15:45:51
```

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

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

4.11.4 Shutting down your system and restarting it

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

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

Perform the following steps to bring the system back up:

- Open an HMC session.
- Select your LPAR.

- Use the circular arrow racetrack buttons to get to the CPC Recovery (or just Recovery) menu.
- Double-click the **Integrated 3270 Console** menu item. A new window should appear.
- Double-click the **LOAD** menu item. The Load Address (**D850** in this example) and Load Parameter (**SYSG**) fields should be correct from the previous IPL.
- Select **Clear**. The Load Address and Load Parameter fields should be correct from the previous IPL. Click **OK**.
- Click **Yes** on the Load Task Confirmation window.
- 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`.
- Press the F10 key to continue the IPL of your z/VM system. This should take 1-3 minutes.
- At the Start prompt you have to specify a **FORCE** start, again because the spool volume label has changed:

```
==> force drain
```

- Do not change the time of day clock:

```
==> no
```

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

```
==> disc
```

- Close the HMC windows.
- Start a 3270 emulator session as the TCPIP service machine should be up. Log on as MAINT.
- Get a 3270 session as MAINT and verify that the volume labels have changed with the **QUERY CPOWNED** command:

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

Important: In the event that you IPLed a system with duplicate system volumes, it is possible that you may have destroyed your saved segments. You will know this is the case when you cannot **IPL CMS**. Rather, you will have to **IPL 190**. To rebuild saved segments, try the following commands (*only do this* if your saved segments are trashed!):

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

4.12 Restoring your z/VM system from tape

It is good to practice to restore a system. You do not want to be doing 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. If you have backed up your system in 4.10, “Backing up your z/VM system to tape” on page 65, 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 chapter describes how to apply the two main types of service:

- A Recommended Service Upgrade (RSU), which is analogous to a Service Pack.
- A Programming Temporary Fix (PTF), which is analogous to a bug fix.

The processes to install these types of service are basically the same.

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

The application of corrective service to z/VM is covered in two manuals:

- *z/VM V6.1 Guide for Automated Installation and Service* (see Part 4), on the web at:

  http://publibz.boulder.ibm.com/epubs/pdf/hcsk2c00.pdf

- *z/VM Service Guide, Version 6, Release 1*, on the web at:

  http://publib.boulder.ibm.com/epubs/pdf/hcsf1c00.pdf

These manuals are much more complete than this chapter. You might consider using these first, rather than this chapter, or you should certainly use them as references.

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 is successfully completed, the PUT2PROD EXEC places the z/VM components, features, or products that are installed on the z/VM System deliverable, and were serviced, into production. A good web site to start at is:

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

The body of the page should look similar to Figure 5-1.

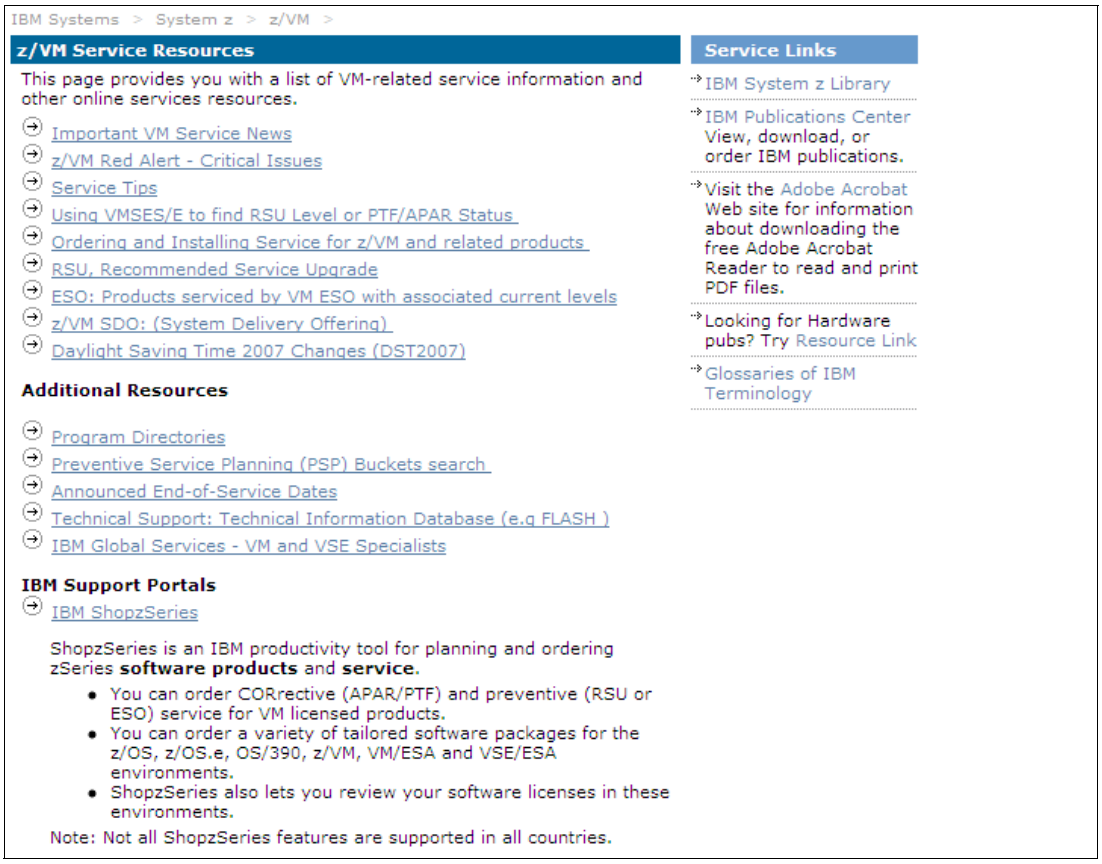


Figure 5-1 z/VM Service main web page

You may want to consider viewing some of the links from this page.

The following sections comprise this chapter:

- “Applying a Recommended Service Upgrade” on page 75
- “PTFs for the zEnterprise 196” on page 82
- “Determining z/VM’s service level” on page 87
- “Applying a PTF” on page 88

5.1 Applying a Recommended Service Upgrade

Applying a Recommended Service Upgrade (RSU) is very similar to applying a PTF described in the previous section. 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. You may prefer to use the official z/VM documentation.

Following is an example of upgrading to a z/VM 6.1 RSU with the medium being files downloaded from the Internet.

The section that follows is a summary of applying service and also describes how to obtain service over the Internet using IBM ShopzSeries.

First determine whether your system needs service. Use the **QUERY CPLEVEL** command:

```
==> q cplevel
z/VM Version 6 Release 1.0, service level 0901 (64-bit)
Generated at 09/11/09 16:51:48 EDT
IPL at 08/31/10 08:44:19 EDT
```

The *service level* (or RSU) is a 4-digit field comprised of two segments, each consisting of two digits. The first two digits represent the last two digits of the year and the second two digits represent the sequential RSU level within that year. Some examples are `0903RSU` and `1002RSU`. With `0903`, the first two digits in the level, `09`, represent the last two digits of the year 2009 and the `03` represents the third RSU service level of that year. Therefore, the `0903` is the third RSU issued in 2009. RSU `1002` would be the second RSU issued in 2010.

The overall steps in applying an RSU are as follow:

- “Getting service from the Internet” on page 75
- “Downloading the service files” on page 76
- “Creating a new MAINT minidisk” on page 77 (**not usually required**)
- “Receiving, applying, and building the service” on page 78
- “Putting the service into production” on page 81

5.1.1 Getting service from the Internet

An RSU is obtained by its PTF number. The PTF for the most current RSU is of the form UM97xyz where xyz is the z/VM version-release-modification level. So for z/VM 6.1 the RSU would be UM97610.

With ShopzSeries, knowing the PTF number is not necessary. If you know you want the latest RSU, you can get it directly, based on the version of z/VM you are running.

Perform the following steps (note that these same steps are documented with some window shots in 5.4, “Applying a PTF” on page 88):

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

  `http://www.vm.ibm.com/service/`
- Click **IBM ShopzSeries** under the IBM Support Portals section.
- Click the link **Sign In for registered users**. If you have a user ID and password, use that. If you do not, click the link **New user registration** and fill out the form to create an ID and password. You must have your IBM customer number.
- Click the link **Create new software orders** at the top.

- The My Orders page should show. Under the Package Category section, click **z/VM - Service** and also choose **RSU recommended service** in the drop-down menu. Click **Continue**.
- There will be five windows of forms that are hopefully self-explanatory. On window 3 of 5, choose the radio button that is applicable to your version of z/VM. In this example it was **z/VM Version 6.1.0 Stacked 6103RSU (PTF UM97610)**.
- On window 4 of 5 choose **Internet** as the delivery mechanism.
- On window 5 of 5, complete the form and click **Submit**.
- In a few minutes, you should get two e-mails, one for the core RSU and one for the PSP bucket (additional fixes that may have come out after the RSU).

5.1.2 Downloading the service files

In this example, the service files are staged on a desktop machine, then copied to z/VM with FTP.

- Download the files to your desktop or another staging system. This example has two files: the `SHIPTFSS` file is for the PSP bucket and the `SHIPRSU1` file is for the RSU.
- FTP the file to the MAINT 500 disk. Following is an example of FTPing from a DOS session:

```
C:\Downloads>ftp 9.60.18.249
User (9.60.18.249:(none)): maint
Password:
ftp> cd maint.500
...
ftp> bin
...
ftp> quote site fix 1024
...
ftp> put S9338801.shiptfss
...
ftp> put S9338766.shiprsu1
...
ftp> quit
```

- Log on to `MAINT`. Access the MAINT 500 disk as file mode `C`. Query the disks:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
MNT191 191  A   R/W   175 3390 4096       41        214-01      31286
31500
MNT5E5 5E5  B   R/W     9 3390 4096      131       1290-80        330
1620
MNT500 500  C   R/W   600 3390 4096        3      38497-36      69503
108000
MNT51D 51D  D   R/W    26 3390 4096      305       1574-34       3106
4680
MNT190 190  S   R/O   100 3390 4096      691      14921-83       3079
18000
MNT19E 19E  Y/S R/O   250 3390 4096     1021      28225-63      16775
45000
```

- Deterse the files.

```
==> deterse s9338801 shiptfss c = servlink =
==> deterse s9338766 shiprsu1 c = servlink =
```

Usually this step should succeed. However, very large RSUs can fill up the MAINT 500 disk either on the **FTP** or the **DETERSE** steps. For example, you may get the error on the **DETERSE** step:

```
DMSERD107S Disk C(500) is full
 No traceback - not enough CTL storage
```

If this occurs, an extra step of creating a new disk is necessary.

5.1.3 Creating a new MAINT minidisk

Important: Normally, this step is not necessary. Some RSUs can be so large that they will not fit on the MAINT 500 minidisk. This is the case with the stacked RSU 5405 for z/VM 5.4.

If you have adequate space to **DETERSE** the files on the MAINT 500 disk, you can skip this section. If you received the error `DMSERD107S Disk C(500) is full` on the previous step, creating a new minidisk for MAINT will be necessary. If so, perform the following steps:

- Create a new MAINT 501 disk for temporary storage of the uncompressed RSU by using 400 cylinders of space taken from the end of the W02 disk (volser is `UV6284` in this example). Verify that the disk layout is good, then bring the changes online with the **DIRECTXA** command:

```
==> acc 2cc c
DMSACC724I 2CC replaces C (500)
==> x user direct c
...
USER MAINT LNX4VM 128M 1000M ABCDEFG
 AUTOLOG AUTOLOG1 OP1 MAINT
 ACCOUNT 1 SYSPROG
...
* add a new MAINT 501 disk for additional space for service files
 MDISK 501 3390 2371 400 UV6284  MR LNX4VM LNX4VM LNX4VM
...
==> diskmap user
...
==> x user diskmap
... // check the report file for gaps or overlaps
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 45 disk pages
```

- Log off `MAINT` and log back on to load the new directory entry. An attempt is made to access the MAINT 500 and 501 disks as file mode `C` and `F`, respectively. However, the new 501 disk has never been formatted. Format it and access it as file mode `F`:

```
==> log
... // log back on
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> acc 501 f
```

```
DMSACP112S F(501) device error
==> format 501 f
DMSFOR603R FORMAT will erase all files on disk F(501). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
mnt501
DMSFOR733I Formatting disk F
```

Now that a new MAINT 501 disk is available, it can be used to stage the RSU file:

- Move the large RSU file from the MAINT 500 (C) to the 501 (F) disk and query the disks:

```
==> copy s8873950 shiprsu1 c = = f
==> erase S8873950 shiprsu1 c
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
MNT191 191  A   R/W   175 3390 4096       41        214-01      31286
31500
MNT5E5 5E5  B   R/W     9 3390 4096      131       1290-80        330
1620
MNT500 500  C   R/W   600 3390 4096        2      13054-12      94946
108000
MNT51D 51D  D   R/W    26 3390 4096      305       1574-34       3106
4680
MNT501 501  F   R/W   400 3390 4096        1      45207-63      26793
72000
...
```

- Deterse the RSU from the 501 disk (F) back to the 500 disk (C) and again query the disks:

```
==> deterse s8873950 shiprsu1 f = servlink c
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
TOTAL
MNT191 191  A   R/W   175 3390 4096       41        214-01      31286
31500
MNT5E5 5E5  B   R/W     9 3390 4096      131       1290-80        330
1620
MNT500 500  C   R/W   600 3390 4096        4      98341-91       9659
108000
MNT51D 51D  D   R/W    26 3390 4096      305       1574-34       3106
4680
MNT501 501  F   R/W   400 3390 4096        1      45207-63      26793
72000
...
```

This shows that the MAINT 500 disk is now 91% full. The tersed file on the 501 disk is no longer necessary, but it is left there for reference.

5.1.4 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 more lengthy and detailed 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 ALL** command. On the other hand, the previous method is more granular and better enables the system administrator to know which pieces of service have been applied.

- Apply the service with the **SERVICE ALL** command. The RSU must be applied first (**S8873950 SERVLINK** in this example). Then any PTFs that came after the RSU can be applied:

```
==> service all S9338766
...
VMFSRV2760I SERVICE processing completed successfully for GCS BUILD
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
Ready; T=129.22/138.98 10:14:11
```

A return code of 0 is ideal. If the last `Ready` line has a number in parenthesis, that is the return code. 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. View details with the **VMFVIEW** command:

```
==> vmfview service
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
No messages meet the search criteria.
********************************************************************************
****             SERVICE              USERID: MAINT                ****
********************************************************************************
****            Date: 09/16/10            Time: 15:45:29            ****
********************************************************************************
* * * End of File * * *
```

You may also see warning messages. For example:

```
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 12 <===> Number of messages not shown = 985
********************************************************************************
****             SERVICE              USERID: MAINT                ****
********************************************************************************
****            Date: 12/17/09            Time: 10:06:17            ****
********************************************************************************
CK:VMFSUI2104I PTF UM32616 contains user information. Review the :UMEMO
CK:            section in file UM32616 $PTFPART
CK:VMFSUI2104I PTF UM32616 contains user information. Review the :UMEMO
CK:            section in file UM32616 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK:            section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK:            section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
```

```
CK:            section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK:            section in file UA46229 $PTFPART
WN:VMFBDC2250W The following OSA objects have been built on BUILD0 100
WN:            (L) and should be copied to your workstation:
WN:VMFBDC2250W IOAJAVA  BIN
CK:VMFSUI2104I PTF UM32501 contains user information. Review the :UMEMO
CK:            section in file UM32501 $PTFPART
CK:VMFSUI2104I PTF UM32654 contains user information. Review the :UMEMO
CK:            section in file UM32654 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (J) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTX EXEBIN
```

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

- Press F3 to get out of **`XEDIT`**.
- Re-IPL CMS and press Enter at the `VM READ` prompt.

```
==> ipl cms
z/VM V5.4.0    2008-10-22 15:36

Ready; T=0.01/0.01 10:46:46
```

- Re-access the MAINT 500 disk as C.

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

- Apply the PSP bucket (***`S9338801`*** in this example):

```
==> service all S9338801
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed with warnings
Ready(00004); T=29.96/33.46 15:55:40
```

In this example, the service was installed, but there were warnings.

- Run the **`VMFVIEW SERVICE`** command:

```
==> vmfview service
    ===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 1 <===> Number of messages not shown = 510
********************************************************************************
****             SERVICE               USERID: MAINT                ****
********************************************************************************
****            Date: 09/16/10            Time: 15:53:09            ****
********************************************************************************
RO:VMFAPP2112W PTF UK59536 has a IFREQ requisite for PTF UM33113 in
RO:            product 6VMCMS10 (CMS component for z/VM 6.1.0)
* * * End of File * * *
```

This message is letting you know that there is a relationship between the two PTFs (`UM33113` and `UK59536`). It is advisable to make sure you have both, or know about the requisite and decide it is not important in your environment.

- Press F3 to get out of **`XEDIT`**.

5.1.5 Putting the service into production

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

Important: The **PUT2PROD** command will affect your production environment. We recommend that all users be logged off before running it. Placing service into production should be performed as part of a planned system outage because a **SHUTDOWN REIPL** is recommended after running it.

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

```
==> put2prod
...
VMFP2P2760I PUT2PROD processing completed successfully for SAVECMS
VMFP2P2760I PUT2PROD processing completed with warnings
Ready(00004); T=13.93/15.21 16:03:13
```

- The return code was 4 in this example. Review the warning messages with the **VMFVIEW PUT2PROD** command:

```
==> vmfview put2prod
    ===> VMFVIEW - Message Log Browse of $VMFP2P  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
No messages meet the search criteria.
********************************************************************************
****             PUT2PROD              USERID: MAINT                ****
********************************************************************************
****            Date: 09/16/10            Time: 16:00:26            ****
********************************************************************************
WN:DTCPRD3043W   File PROFILE STCPIP I has been updated; Its content
WN:            should be reviewed for changes that may affect your use of
WN:            this file
WN:DTCPRD3043W   File SCEXIT SAMPASM I has been updated; Its content
WN:            should be reviewed for changes that may affect your use of
WN:            this file
WN:DTCPRD3021W TCP2PROD processing completed with RC = 4
```

  With these warnings you should do as message DTCPRD3043W suggests and compare the files to see whether you need to pick up any of the new changes in your running copy of the sample file.

- Press F3 to get out of **XEDIT**.
- Even though the service has been “put into production”, the **QUERY CPLEVEL** command should still return the current service level, in this example 0901. This is because the new CP load module (nucleus) has not been loaded:

```
==> q cplevel
z/VM Version 6 Release 1.0, service level 0901 (64-bit)
Generated at 09/11/09 16:51:48 EDT
IPL at 09/15/10 15:52:34 EDT
```

- To load 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 0903:

```
==> shutdown reipl iplparms cons=sysc
HCPSHU960I System shutdown may be delayed for up to 330 seconds
Ready; T=0.01/0.01 11:12:32
```

- After the system comes back up in a few minutes, start a new 3270 session and log on as MAINT.
- Run the **QUERY CPLEVEL** command again:

```
==> q cplevel
z/VM Version 6 Release 1.0, service level 1002 (64-bit)
Generated at 09/16/10 15:54:07 EDT
IPL at 09/16/10 16:07:01 EDT
```

This shows that the new CP load module is now being used, and that the service level is the second RSU in the year 2010.

5.2 PTFs for the zEnterprise 196

In September of 2010, a new mainframe became available: the zEnterprise 196. See the following web site for a list of the PMRs that apply to it:

http://www.vm.ibm.com/service/vmreqze.html

This web page also includes a link to the Preventative Service Planning (PSP) bucket for z/VM on the zEnterprise 196. The PSP bucket should always contain all the latest service information for z/VM on the z196.

Table 5-1 shows a summary of the APARS for z/VM 6.1.

Important: This list was correct at the time of the writing of this book in late 2010. It could change, so refer to the previous web page to confirm. Also, it is likely that all of the PTFs associated with these APARs will be rolled into the first RSU of 2011. So if you are up to service level 1101 or later, you can verify that the PTFs are applied with the steps shown in 5.2.3, “Verifying that the zEnterprise 196 service is applied” on page 86.

Table 5-1 z/VM 6.1 APARs for the zEnterprise 196

APAR	Component	Description
VM64774	CP	Set/Query reorder command
VM64798	CP	zEnterprise 196 Processor Support
VM64879	CP	zEnterprise 196 Processor Support
VM64881	CP	VM Coupling Facility hang at IPL
VM64793	CP	Secure-Key Bulk Encryption Support
VM64820	PERFTK	New function in the Performance Toolkit
VM64814	CP	XRC Time-stamping Support
VM64807	EREP	EREP support for zEnterprise 196
VM64672	HCD	HCD support for zEnterprise 196
VM64747	HCM	HCM support for zEnterprise 196
VM64799	CMS	IOCP support for zEnterprise 196

APAR	Component	Description
VM64774	CP	Set/Query reorder command
VM64891	CP	HIPER data corruption issue in VM64709, EAV support

Because support for HCD and HCM was not necessary for the system used in the examples in this book, only the PTFs for the following APARs were ordered from ShopzSeries:

```
VM64774 VM64798 VM64879 VM64881 VM64793 VM64820 VM64814 VM64807 VM64799 VM64818
VM64891
```

5.2.1 Ordering service for the zEnterprise 196 PTFs

This section briefly describes how to order PTFs for the zEnterprize 196. Perform the following steps.

- Follow the steps in 5.1.1, “Getting service from the Internet” on page 75, up to the point where you click **z/VM - Service** on the *My orders* page.
- Rather than clicking RSU Recommended Service Upgrade in the drop-down menu to the right, accept the default of **Individual PTFs**. Click **Continue**.
- In Step 1 of 5, select **Individual PTFs by APAR number** as shown in Figure 5-2. Click **Continue**.

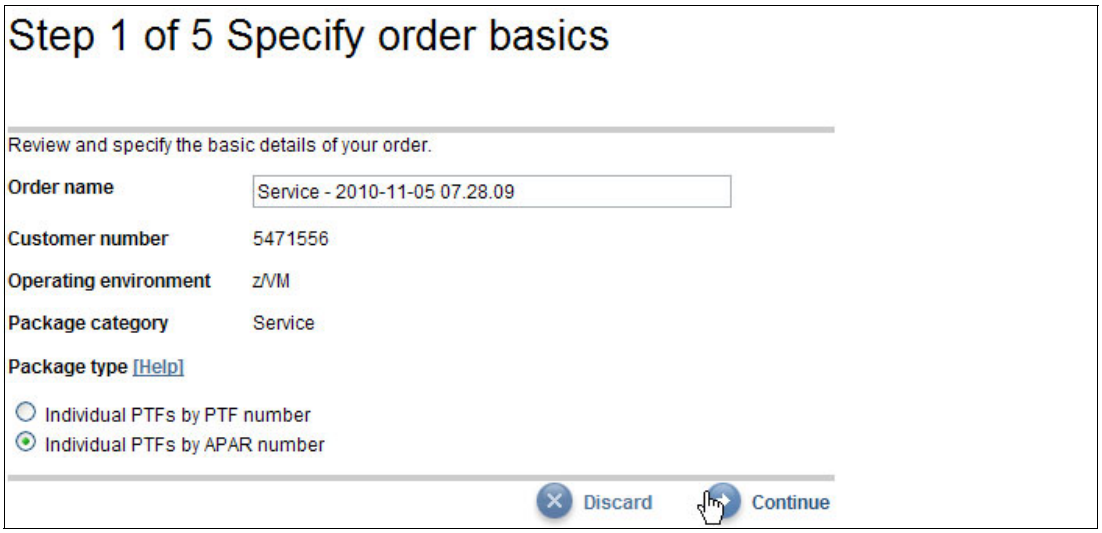


Figure 5-2 Ordering PTFs by APAR number

- In Step 2 of 5, accept the default of **Do not use a report for this order** and click **Continue**.
- In Step 3 of 5, enter the APAR numbers as shown in Figure 5-3 on page 84.

Figure 5-3 Specifying service order contents

- In Step 4 of 5, specify your delivery options. In this example, **Internet** was chosen as the preferred media, and no alternate method was chosen. Click **Continue**.
- In Step 5 of 5, review your order and click **Submit** when it is correct.
- You can leave that web page up and click **Refresh order status** from time to time. It should move from Submitted to Received to Final Packaging to becoming a link named `Download`.
- Click **Download** when it becomes available. You should see a window similar to what is shown in Figure 5-4.

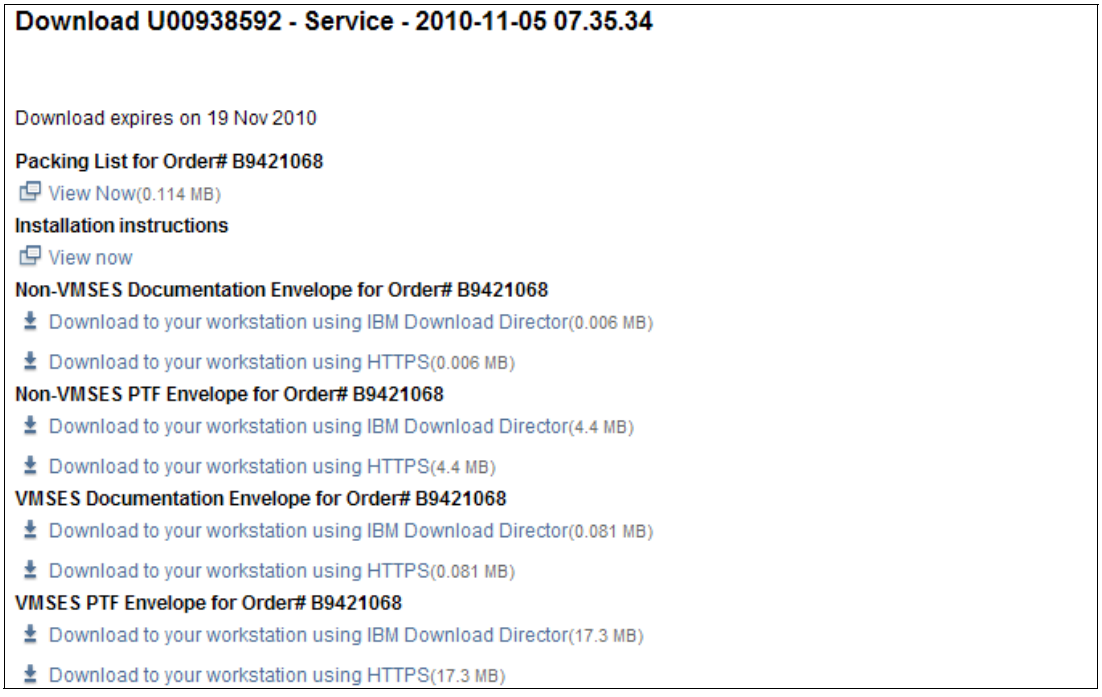


Figure 5-4 Downloading service for zEnterprise 196 PTFs

- Download the two documentation envelopes and the two PTF envelopes to your workstation or other staging system.

- Complete the steps in a similar fashion to those starting at 5.1.2, “Downloading the service files” on page 76. This completes the process of applying the SES PTFs (with file types ending in S).
- Refer to the following section to apply the non-SES PTF (with file types ending in N).

You may consider doing a **SHUTDOWN REIPL** at this point, or wait until after the next section.

5.2.2 Applying the non-SES PTF UV61111

At the time of the writing of this book, PTF UV61111 corresponded to APAR VM64807. This PTF is non-SES, which means it cannot be applied using the typical **SERVICE ALL** and **PUT2PROD** commands.

- After you get the PTF from ShopzSeries, copy it to the MAINT 500 disk in binary fixed 1024 byte record format. In the previous example, four files with a file name of S9421068 were uploaded to the MAINT 500 disk. The one with a file type of SHIPTFSS was **DETERSE**d to a new file type of SERVLINK and applied with **SERVICE ALL** and **PUT2PROD**.
- Access the MAINT 500 disk as C:

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

- List the files that you uploaded. In this example, the file name is ***S9421068***:

```
==> filel S9421068 * c
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
      S9421068 SERVLINK C1 V       4005      18865      14243 11/05/10 13:52:19
      S9421068 SHIPTFSS C1 F       1024      17686       4422 11/05/10 13:04:43
      S9421068 SHIPTFSN C1 F       1024       4466       1117 11/05/10 13:04:37
      S9421068 SHIPDOCS C1 F       1024         83         21 11/05/10 13:04:28
      S9421068 SHIPDOCN C1 F       1024          6          2 11/05/10 13:04:25
```

  The two files in bold are non-SES, identified by a trailing N.

- Deterse the object code file to a file with a type of NOSESLNK and the documentation file to a file with a type of NOSESDOC. This can be done directly from **FILELIST** with the following **DETERSE** commands:

```
      S9421068 SERVLINK C1 V       4005      18865      14243 11/05/10 13:52:19
      S9421068 SHIPTFSS C1 F       1024      17686       4422 11/05/10 13:04:43
deterse / = noseslnk =  C1 F       1024       4466       1117 11/05/10 13:04:37
      S9421068 SHIPDOCS C1 F       1024         83         21 11/05/10 13:04:28
deterse / = nosesdoc =  C1 F       1024          6          2 11/05/10 13:04:25
```

- Press F3 to get out of **FILELIST**.
- Perform the following **VMFPLCD** command:

```
==> vmfplcd scan env= s9421068 noseslnk c (disk date eod
```

- This should create the file DISK MAP on your A disk. Edit the file and view the lines with:

```
==> x disk map
====> pre off
====> ALL /ERPTFLIB
 ERPTFLIB TLB61111 U1               F         80      22266  08/24/10 16:46:32
 ERPTFLIB TLB60820 U1               F         80      21911  09/29/03 20:02:53
 ERPTFLIB TLB60786 U1               F         80      21882  03/26/03 16:57:52
 ERPTFLIB TLB60432 U1               F         80      21791  06/01/99 09:18:46
 ERPTFLIB TLB60345 U1               F         80      19312  12/10/98 11:28:23
```

Note that the most recent file has a date of 2010 and the last five digits of the file type correspond to the last five digits of the PTF.

- The EREP program directory states that just one file needs be copied. Perform the following **VMPLCD** commands to do this:

```
==> vmfplcd rst
==> vmfplcd load erptflib tlb61111 a (eod
Loading ...
 End-Of-Group OR End-Of-Disk
 ERPTFLIB TLB61111 A1
```

- Access the MAINT 201 disk as file mode Z, back up the old EREP TXTLIB and replace it with the new one on the A disk:

```
==> acc 201 z
==> rename erptflib txtlib z erptflib tlbold z
==> copy erptflib tlb61111 a erptflib txtlib z (replace
```

- A **SHUTDOWN REIPL** is not necessary. However, if you did not do one in the previous section, one is recommended now. Otherwise, the EREP virtual machine can just be recycled with the **FORCE** and **XAUTOLOG** commands:

```
==> force erep
USER DSC   LOGOFF AS  EREP     USERS = 11    FORCED BY MAINT
==> xautolog erep
Command accepted
AUTO LOGON  ***       EREP     USERS = 12
HCPCLS6056I XAUTOLOG information for EREP: The IPL command is verified by the
IP
L command processor.
```

You should now have all the service needed for the zEnterprise 196.

5.2.3 Verifying that the zEnterprise 196 service is applied

A short REXX EXEC is written and run to verify that service for the zEnterprise 196 has been applied:

```
==> type check910 exec

/* EXEC to check for z196 PTFs */
'service cp status VM64774'
'service cp status VM64798'
'service cp status VM64879'
'service cp status VM64881'
'service cp status VM64793'
'service perftk status VM64820'
'service cp status VM64814'
'service cms status VM64799'
'service cp status VM64818'

==> check910
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (6VMCPR10%CP) APAR VM64774 (PTF UM33169) status:
VMFSRV1226I    RECEIVED  11/05/10 13:52:51
VMFSRV1226I    APPLIED   11/05/10 13:52:52
VMFSRV1226I    BUILT     11/05/10 13:53:57
VMFSRV1226I    PUT2PROD  11/05/10 13:55:55
```

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

Verify that all of the APARs are reported as received, applied, built, and put into production.

5.3 Determining z/VM's service level

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

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

Perform the following steps:

- Log on to TCPMAINT. Use the **QUERY VMLAN** command to determine the latest APAR applied:

```
==> cp query vmlan
VMLAN maintenance level:
  Latest Service: VM64604
VMLAN MAC address assignment:
  MACADDR Prefix: 020003
  MACIDRANGE SYSTEM: 000001-FFFFFF
             USER:   000000-000000
VMLAN default accounting status:
  SYSTEM Accounting: OFF       USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 1
  TRANSIENT  Limit: INFINITE   Current: 0
```

  This shows that the latest APAR applied is VM64604.

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

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-03-00-00-01
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
  RDEV: 1004.P00 VDEV: 1004 Controller: DTCVSW1
  RDEV: 1100.P00 VDEV: 1100 Controller: DTCVSW2  BACKUP
```

  This shows the controller is named DTCVSW1.

- Use the **NETSTAT** command with the controller name to determine the maintenance of the TCPIP MODULE:

```
==> netstat tcp dtcvsw1 level
VM TCP/IP Netstat Level 540       TCP/IP Server Name: DTCVSW1

IBM 2084; z/VM Version 5 Release 4.0, service level 0903 (64-bit), VM TCP/IP
Lev
el 540; RSU 0903 running TCPIP MODULE E2 dated 12/17/09 at 10:53
TCP/IP Module Load Address:  00C21000
```

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

```
==> tcpslvl tcpip module e
DTCLVL3306I SLVL data obtained; file TCPIP SLVLDATA A created
==> x TCPIP SLVLDATA
SLVL  TCPIP PK67610
...
SLVL  TCTOOSD PK98608
...
```

5.4 Applying a PTF

You may determine that you need to apply a specific fix or PTF to your system. For example, an Authorized Program Analysis Report (APAR), VM64670, was opened when Linux guests were hanging intermittently. The summary of the APAR is as follows:

```
PROBLEM SUMMARY:     LINUX USER HUNG BECAUSE SVPBK LOCK HELD
USERS AFFECTED:      All users of z/VM running Linux guests.
PROBLEM DESCRIPTION: Linux guests may become hung due to a problem in managing
a lock word.  This problem is timing-related and may occur intermittently.
PROBLEM CONCLUSION:  Lock word processing in HCPWED is updated to properly
handle all possible states of the lock.
```

The APAR was assigned the following Programming Temporary Fix (PTF) numbers for each of the following z/VM releases:

z/VM 5.3	UM32809
z/VM 5.4	UM32810
z/VM 6.1	**UM32811**

So for z/VM 6.1, you want to apply PTF UM32811. Following is an example of how to do so.

5.4.1 Getting service using ShopzSeries

Service for z/VM is still available on the media of tape. However, getting service over the Internet is more convenient and becoming more common. Typically this is done with IBM ShopzSeries. Perform the following steps:

- Click the link **IBM ShopzSeries** under the IBM Support Portals heading on the main Service page, as shown in Figure 5-1 on page 74. This should take you to the following address:

  https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp

- From there you can search for an APAR if you have the APAR number. In Figure 5-5 on page 89, the first three steps to do this are shown:
  - On the menu bar at the top, click **Support and Downloads**, then choose **Search** in the drop-down menu. This is shown at the top of the figure.
  - In the Support type menu, choose **System z** and in the Search text area, type the APAR number, VM64670 in this example. This is shown in the middle of the figure.
  - If the APAR is found, you should see a link as a result. Click that link, **VM64670: LINUX USER HUNG**..., in this example. This is shown at the bottom of the figure.

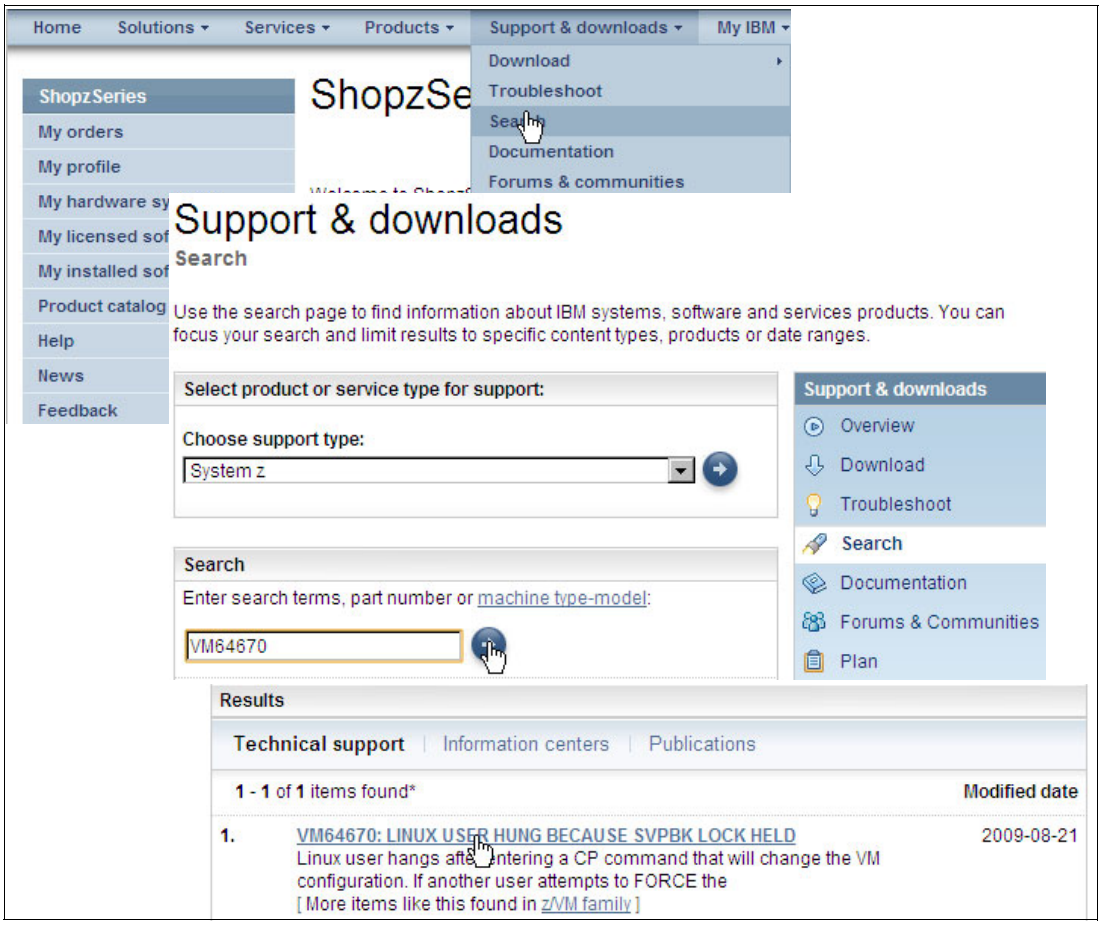
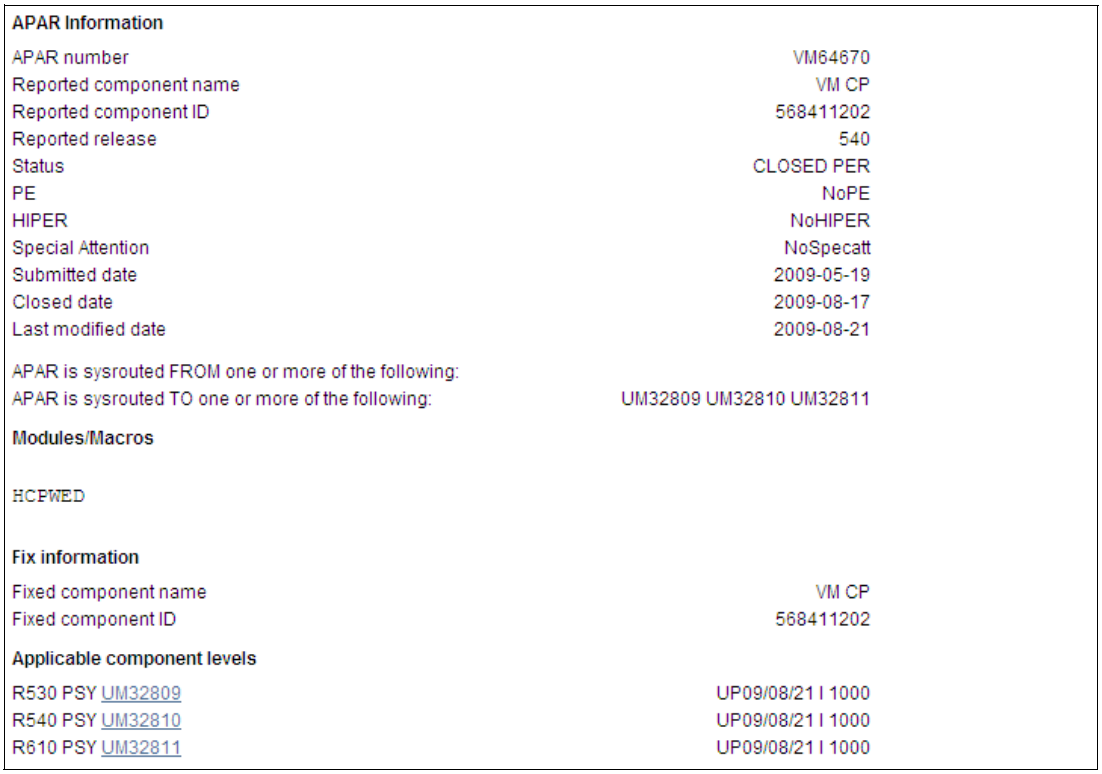


Figure 5-5 Searching for a PTF on ShopzSeries

Clicking the link should bring you to the APAR. In this example, you should find the information on APAR VM64670 that was summarized previously. At the top of the page, look for the section “A fix is available.” In this example, there is a fix available.

Farther down the page, note the Fixed component name, which is important. In this example it is `VM CP` shown near the bottom of Figure 5-6 on page 90.



APAR Information

APAR number	VM64670
Reported component name	VM CP
Reported component ID	568411202
Reported release	540
Status	CLOSED PER
PE	NoPE
HIPER	NoHIPER
Special Attention	NoSpecatt
Submitted date	2009-05-19
Closed date	2009-08-17
Last modified date	2009-08-21
APAR is sysrouted FROM one or more of the following:	
APAR is sysrouted TO one or more of the following:	UM32809 UM32810 UM32811

Modules/Macros

HCPWED

Fix information

Fixed component name	VM CP
Fixed component ID	568411202

Applicable component levels

R530 PSY UM32809	UP09/08/21 I 1000
R540 PSY UM32810	UP09/08/21 I 1000
R610 PSY UM32811	UP09/08/21 I 1000

Figure 5-6 Web page for APAR VM64670

At the bottom of the page the “Applicable component levels” section shows that PTF UM32811 is available for z/VM 6.1. Before getting that PTF, you may want to be sure that it has not already been applied.

5.4.2 Determining whether a PTF has been applied

Check to make sure that the PTF has not previously been applied. In this example it is known that the PTF is UM32811 and the component is VM CP.

- Because the description of the PTF cites a component name of VM CP, the component CP is used in the following command.
- Use the **SERVICE** command to query whether the PTF has been applied:

```
==> service cp status um32811
VMFSRV2760I SERVICE processing started
VMFSRV1227I UM32811 is not received or applied to CP (6VMCPR10%CP)
VMFSRV2760I SERVICE processing completed successfully
```

This shows that PTF UM32811 has *not* been applied. The sections that follow describe how to obtain and apply it.

5.4.3 Downloading the service to z/VM

From the previous APAR web page search, the link for UM32811 is clicked, which results in a web page that should be similar to the one shown in Figure 5-7 on page 91.

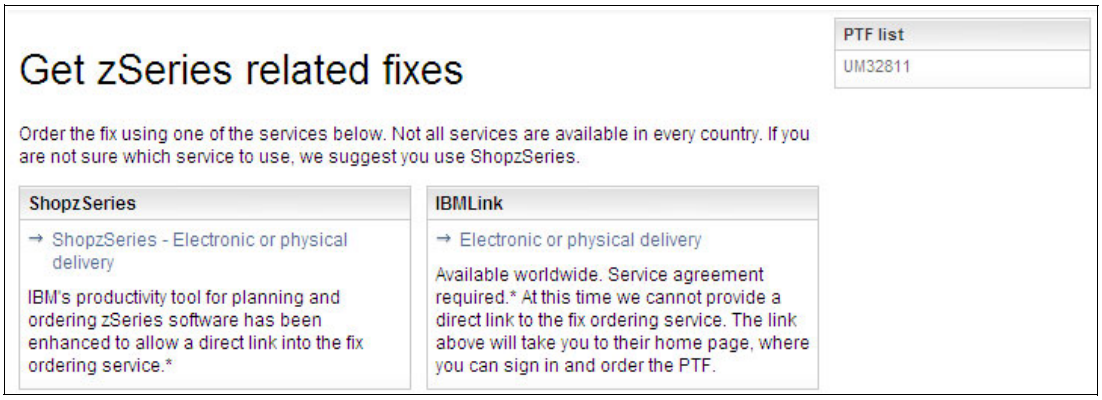


Figure 5-7 Getting fixes from ShopzSeries on IBMLink

- In this example, the link **ShopzSeries - Electronic or physical delivery** is selected. Sign into ShopzSeries with your IBM ID and follow the five self-explanatory steps to order your PTF. When you are finished, click **Submit** to place your order.
- You should receive an e-mail within a few minutes. It will have your order number and a link to start the download of service files. Following is an example of the important information in the e-mail:

```
From:    Oms Client01/Boulder/IBM
Subject: IBM Order <Bxxxxxxx> is ready for download.
...
To access your order directly, go to:
https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp?action=downlo
ad&orderId=<Uxxxxxxxd>0
```

- Point your browser to the link in the e-mail. You should see a web page similar to the one shown in Figure 5-8.

Figure 5-8 Web page created for downloading a PTF

- Choose a method of downloading the VMSES PTF Envelope for your order to your desktop machine. You may also choose to download the VMSES Documentation Envelope.
- There should be a SES envelope (the PTF or PTFs themselves) and a documentation envelope. Copy both to z/VM in binary with fixed 1024-byte records to the MAINT 500 disk. Usually, FTP is used. The PTF 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 of FTPing from a DOS session:

```
C:\downloads> ftp 9.60.18.249
User (9.60.18.249:(none)): maint
Password:
...
ftp> cd maint.500
...
ftp> bin
...
ftp> quote site fix 1024
...
ftp> mput s8873674.*
mput S8873674.SHIPDOCS? y
...
ftp: 6144 bytes sent in 0.05Seconds 130.72Kbytes/sec.
mput S8873674.SHIPTFSS? y
...
ftp: 4096 bytes sent in 0.01Seconds 273.07Kbytes/sec.
ftp> quit
```

- Log on to z/VM as `MAINT`.
- Access the MAINT 500 disk as `C`:

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

- The envelope files arrive in a compressed format to speed downloads. In order to use them they must first be renamed to have a file type of `SERVLINK` and uncompressed with the **`DETERSE`** command. Therefore, we recommend to leave the file name of the SES envelope unchanged, but to change the prefix letter of the documentation envelope to `D`. First rename them, then use the **`DETERSE`** command with the **`(REPLACE`** parameter to uncompress them in place and save disk space:

```
==> rename s8873674 shipftss c = servlink =
==> rename s8873674 shipdocs c d8873674 servlink =
==> deterse s8873674 servlink c = = = (replace
==> deterse d8873674 servlink c = = = (replace
```

  Be sure all commands complete successfully.

5.4.4 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 minidisk with a lot of free space—that is what the MAINT 500 minidisk is for.

- Access the MAINT 500 disk as file mode C:

```
==> acc 500 c
```

```
DMSACC724I 500 replaces C (2CC)
```

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

```
==> service all d8873674
...
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
==> service all s8873674
...
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

  If you see no number in parenthesis after the `Ready;` prompt, then the return code is 0. Any non-zero return code will be in parenthesis. A return code of 0 is ideal. In general a return code of 4 is acceptable. It means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

- The output files are of the form `$VMF* $MSGLOG`. You may wish to inspect these files.

```
==> filel $vmf* $msglog
      $VMFSRV  $MSGLOG  A1 V         80        728         14 12/15/09
13:43:34
      $VMFBLD  $MSGLOG  A1 V         80        787         11 12/15/09
13:41:47
      $VMFAPP  $MSGLOG  A1 V         80        252          4 12/15/09
13:41:37
      $VMFREC  $MSGLOG  A1 V         80         56          1 12/15/09
13:41:36
      $VMFMRD  $MSGLOG  A1 V         80        231          4 12/15/09
13:41:35
      $VMFP2P  $MSGLOG  A1 V         80        805         15 11/19/09
13:52:09
      $VMFINS  $MSGLOG  A1 V         80        163          3 11/19/09
13:47:25
```

- 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
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 1 <===> Number of messages not shown = 3
********************************************************************************
****             SERVICE              USERID: MAINT                ****
********************************************************************************
****            Date: 12/15/09            Time: 13:43:34            ****
********************************************************************************
====> F3
```

Ideally there will be no output. If there are errors, they must be addressed. If there are warnings, they may be acceptable but should be investigated.

5.4.5 Putting the service into production

To put the service into production, perform the following steps:

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

```
==> put2prod
...
VMFP2P2760I PUT2PROD processing completed successfully
```

  Again, watch for a return code of 0.

- Your PTF should now be *put into production*. You may or may not have to re-IPL the system, depending on the nature of the PTF applied. If you are in a position to re-IPL your system, it may be safest to re-IPL using 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. When the system comes back, start a 3270 session to MAINT and again query the status of the PTF:

```
==> service cp status um32811
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (6VMCPR10%CP) PTF UM32811 status:
VMFSRV1226I    RECEIVED  12/15/09 13:41:36
VMFSRV1226I    APPLIED   12/15/09 13:41:37
VMFSRV1226I    BUILT     12/15/09 13:42:14
VMFSRV1226I    PUT2PROD  12/15/09 13:47:59
VMFSRV2760I SERVICE processing completed successfully
```

This shows that the PTF has been successfully applied.

5.4.6 Checking for APARMEMO files

After you have applied the PTFs, you should check for files with a file type of APARMEMO on the MAINT 500 disk. These files may have additional instructions on work to do after the PTFs have been applied. Perform the following steps:

- Access the MAINT 500 disk as C and list the files with file type APARMEMO:

```
==> acc 500 c
==> listfile * aparmemo c
6VMCMS10 APARMEMO C1
```

  In this example, there is one APARMEMO file.

- Look at the contents of the file:

```
==> type 6vmcms10 aparmemo c

APAR MEMOS      01/26/10.12:50:20
================================

        THE FOLLOWING MEMOS WERE INCLUDED WITH THE PTFS SHIPPED:

        NONE.
```

In this example, the APARMEMO file was created, but no additional memorandums are present.

You will not see any new information in the APARMEMO file if you have not done **SERVICE** against the documentation SERVLINK file. This is because the <prodid> MEMO file is in the documentation SERVLINK file.

5.5 Moving on

You should now be done installing, configuring and servicing 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 for SLES 11 SP1

There are several possible ways to install SLES 11 SP1 onto the mainframe. It can be installed from an FCP attached DVD-ROM (in this case all data are read directly from DVD) or from network sources like NFS, FTP, HTTP and SMB. Installation using an NFS server is used in this book. To accomplish this, it is recommended that you set up a PC Linux system. This server will supply both the SLES 11 SP1 distribution and the files associated with this book.

It must have at least 3 GB of free disk space for one SLES 11 SP1 install server. 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 Linux PC as the NFS server.

You can also choose to use a Windows workstation using SMB, but this option is not addressed in this book. Often more problems are encountered when using a Windows workstation to serve a SLES installation so this option is not recommended. If you have no other choice, refer to Section 4.2.1, “Making the Installation data Available” in the manual *SUSE Linux Enterprise Server Deployment Guide*. This manual is included on the SuSE DVDs.

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

6.1 Downloading files associated with this book

This book has several 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/SG247931/SG24-7931-00.tgz

The tar file `virt-cookbook-S11SP1.tgz` is only about 20 KB. Download the file and untar it. The following example shows this being done from the directory `/nfs/`:

```
# mkdir -p /nfs
# cd /nfs
... download or copy the file virt-cookbook-S11SP1.tgz ...
# tar xzf SG24-7931-01.tgz
```

This should create a new directory named `virt-cookbook-S11SP1/`. List the files:

```
# ls -F virt-cookbook-S11SP1/
README.txt  clone.sh*  vm/
```

There is a `README` file, a clone script, and a directory for z/VM files. You now have downloaded and untarred the files associated with this book.

Make a directory, `/nfs/sles11sp1/`. This will hold the SLES 11 SP1 install ISO images:

```
# mkdir sles11sp1
```

6.2 Setting up an SLES 11 SP1 install server

You may have a licensed version of SLES 11 SP1 on physical DVDs or you may choose to try an evaluation copy. There is an evaluation copy on the web starting at:

http://www.novell.com/products/server/eval.html

Follow the link named SUSE Linux Enterprise Server 11 Service Pack 1 for System z and create an account to download the ISO images.

6.2.1 SLES 11 SP1 DVD ISO image file

Table 6-1shows the DVD ISO images for SLES 11 SP1 that were available at the time this book was written.

Table 6-1 SLES 11 SP1 DVDs

DVD number	File name	File size in bytes
1	SLES-11-SP1-DVD-s390x-GM-DVD1.iso	2,970,603,520
2	SLES-11-SP1-DVD-s390x-GM-DVD2.iso	4,532,912,128

The second DVD is not described in this book because it is not needed. If you are starting with a soft copy of the DVD 1 ISO image, copy it to the `/nfs/sles11sp1/` directory now.

6.2.2 Starting from a physical DVD

NOTE: Perform this section *only if* you are starting from physical DVD discs.

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

Put the first DVD in the drive. It is often available as the file `/dev/cdrom`. If there is no such file on your system, you will have to determine which file (such as `/dev/hdc` or under the `/media` directory) is the device file for the DVD drive.

Now copy the contents of the DVD to an ISO image using the **dd if** (input file) and **of** (output file) parameters. The following example copies the SLES 11 SP1 DVD to the appropriately named file. Sometimes, `/dev/cdrom` is automatically mounted over `/mnt/cdrom` when you put the DVD in the drive. If so, you will need to unmount it using the **umount** command after copying the contents of the DVD:

```
# cd /nfs/sles11sp1
# dd if=/dev/cdrom of=SLES-11-SP1-DVD-s390x-GM-DVD1.iso
```

The DVD should start spinning; copying will take a couple of minutes. Repeat the **dd** command for each additional disc you have, using appropriate file names.

```
# umount /mnt/cdrom
# mount ... // next DVD
```

6.2.3 Verifying the ISO images

You should first 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 DVD. After downloading or **dd**ing the ISO images, the checksums are calculated again and compared against the original values using the **md5sum** command and the checksum files.

Following is an example of using the **md5sum** command against an `MD5SUM` file. When you obtain the DVDs, be they physical discs or ISO images, you should also obtain an `MD5SUM` file.

SLES 11 SP1 MD5SUM values

```
# cat MD5SUM
71819d092e6177356f85e7d6f2d4085d  SLES-11-SP1-DVD-s390x-GM-DVD1.iso
fe8bc97da097a77c7cb85d0f8d24c400  SLES-11-SP1-DVD-s390x-GM-DVD2.iso
```

Use the **md5sum -c** command to verify the integrity of the ISO images. All should report OK. Following is an example:

```
# md5sum -c MD5SUM
SLES-11-SP1-DVD-s390x-GM-DVD1.iso: OK
SLES-11-SP1-DVD-s390x-GM-DVD2.iso: OK
```

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

6.2.4 Configuring the SLES 11 SP1 install server

With SLES 9 and earlier, creating an install tree became complicated with Service Packs. SLES 10 brought the ability to automount the DVDs. There are different possible methods of making the SLES 11 SP1 distribution available:

- By copying the contents of all DVDs into one directory
- By loop mounting each ISO image to a different directory
- By using the ISO images directly

The last option is the easiest. The installation process will access the ISO images directly and mount them automatically.

The initial RAMdisk (`initrd`) and the kernel (`vmrdr.ikr`), will have to be copied to z/VM for the first Linux install. These files are on the first DVD in the `/boot/s390x/` directory.

To allow access to these files, the first DVD must be loopback mounted using the **`-o loop`** flag to the **`mount`** command. First create a directory, `dvd1/`, as a mount point:

```
# cd /nfs/sles11sp1
# mkdir dvd1
# mount -o loop SLES-11-SP1-DVD-s390x-GM-DVD1.iso dvd1
```

You should now have a directory, `/nfs/`, with subdirectories of `sles11/` and `virt-cookbook-S11SP1/`.

The next step is to enable the NFS server.

6.3 Enabling the NFS server

The method of enabling an NFS server differs depending upon the operating system. However, the steps are basically the same:

- Be sure the NFS RPMs are installed.
- Export the appropriate directories.
- Start the NFS server in the current run level.

Be sure the NFS server is installed. Typically, the RPM is named `nfs-kernel-server` or `nfs-utils`. If this RPM is not installed, install it now.

The directories to export using NFS are set in the `/etc/exports` configuration file. Make a backup copy of the file. Then edit the original copy and add the two directories to be exported:

```
# cd /etc
# cp exports exports.orig
# vi exports       // add two lines at the bottom
...
/nfs/virt-cookbook-S11SP1       *(ro,sync,no_subtree_check)
/nfs/sles11sp1                  *(ro,sync,no_subtree_check)
```

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

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

```
# chkconfig --list 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 will need to set it to run with the **`chkconfig command`** and turn it on for the current run level with the **`rcnfsserver start`** command (if the NFS server is already running, you can use the **`restart`** parameter):

```
# chkconfig nfsserver on
# service nfsserver start
```

```
Starting kernel based NFS server
                                                          done
```

Your NFS server should now be running with the directory exported. It is recommended that you test this by mounting the exported directory locally. The following example shows that the `/mnt/` directory is empty. Then the newly exported `/nfs/` directory is mounted using the `localhost` keyword, and the files are listed.

```
# ls /mnt
# mount localhost:/nfs/sles11sp1/dvd1 /mnt
# ls -F /mnt
AARCHIVES.gz   README          gpg-pubkey-0dfb3188-41ed929b.asc  license.tar.gz
COPYING       boot/           gpg-pubkey-1d061a62-4bd70bfa.asc  ls-lR.gz
COPYING.de    content         gpg-pubkey-307e3d54-4be01a65.asc  media.1/
COPYRIGHT     content.asc     gpg-pubkey-3d25d3d9-36e12d04.asc  pubring.gpg
COPYRIGHT.de  content.key     gpg-pubkey-7e2e3b05-4be037ca.asc  suse/
ChangeLog     control.xml     gpg-pubkey-9c800aca-4be01999.asc  suse.ins
INDEX.gz      directory.yast  gpg-pubkey-a1912208-446a0899.asc
NEWS          docu/           gpg-pubkey-b37b98a9-4be01a1a.asc
```

This shows that the SLES 11 SP1 install root directory is accessible. Now unmount it and test the `virt-cookbook-S11SP1/` directory:

```
# umount /mnt
# mount localhost:/nfs/virt-cookbook-S11SP1 /mnt
# ls -F /mnt
README.txt  clone.sh*  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 files to System z Linux.

6.4 Configuring an FTP server for z/VM installation

This section assumes that you have access to the z/VM 6.1 install code in electronic format. Ordering it through ShopzSeries is briefly described in 4.1.1, “Obtaining z/VM through electronic download” on page 28. If you have completed that section, you may have the two z/VM product install files staged on an intermediate workstation, or you may be ready to download them from the Internet.

6.4.1 Preparing the z/VM product install files

The two zip files correspond to the larger first z/VM product DVD, and to the smaller second DVD,the RSU. The contents of these files must be copied to the directory of the FTP server. To accomplish this, perform the following steps:

- Create a target directory. In this example the directory `/ftp/zvm61/` is used:

  ```
  # mkdir -p /ftp/zvm61
  ```

- Set the group ownership of this directory, recursively, to **`ftp`**. This will allow the FTP daemon, which runs as the user `ftp`, to change directory into it:

  ```
  # chgrp -R ftp /nfs/zvm61
  ```

- Either upload the two z/VM installation zip files from the intermediate workstation, or download them directly from the Internet. The following example shows copying them from

an intermediate workstation Windows DOS session to the FTP server at the IP address 9.60.18.233 in the directory, `/ftp/zvm61/`, using the add-on **pscp** command (Putty scp):

```
C:>pscp *.zip root@9.60.18.233:/ftp/zvm61
...
cd813250.zip              | 1247495 kB | 303.2 kB/s | ETA: 00:00:00 | 100%
CD813270.ZIP              | 44031 kB | 352.3 kB/s | ETA: 00:00:00 | 100%
```

- List the newly copied files:

```
# cd /ftp/zvm61
# ls -l
total 1291532
-rw-r--r--. 1 root root 1277435798 Nov 11 14:08 cd813250.zip
-rw-r--r--. 1 root root   45088210 Nov 11 14:06 CD813270.ZIP
```

- Unzip the files from DVD1, the larger file, using the **unzip** command. This creates the directory `cpdvd/`:

```
# unzip cd813250.zip
Archive:  cd813250.zip
   creating: cpdvd/
  inflating: cpdvd/610GANUC
  inflating: cpdvd/610GARAM
...
```

- Unzip the files from the RSU DVD2, the smaller file. When prompted to replace files, respond with `A` for all:

```
# unzip CD813270.ZIP
Archive:  CD813270.ZIP
  inflating: cpdvd/610rsu.dvdimage
  inflating: cpdvd/61ckdrsu.srl
  inflating: cpdvd/61fbarsu.srl
replace cpdvd/CKD50000? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: cpdvd/CKD50000
  ...
```

You should now have all the z/VM product install files in place under the directory `/ftp/zvm61/cpdvd/`.

6.4.2 Installing and configuring the FTP server

An FTP server must be installed and configured. The vsftpd FTP server is recommended. This section shows how to configure it as an anonymous FTP server. To accomplish these tasks, perform the following steps:

- Use the **rpm -qa** command to see if the RPM is installed:

```
# rpm -qa | grep ftpd
```

- No output shows that it is not installed. Use the **yum -y** command to install the package:

```
# yum -y install vsftpd
Loaded plugins: rhnplugin
This system is not registered with RHN.
...
Installed:
  vsftpd.s390x 0:2.2.2-6.el6
```

- Make a backup of the vsftpd configuration file, `/etc/vsftpd/vsftpd.conf`:

```
# cd /etc/vsftpd
# cp vsftpd.conf vsftpd.conf.orig
```

- Modify the configuration file to set the directory that the anonymous user will be logged in to `/ftp/zvm61/` using the `anon_root` variable. Also disable local (non-anonymous) logins by commenting out the `local_enable=YES` and `write_enable=YES` lines.

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
# set the home directory of anonymous FTP to /ftp/zvm61
anon_root=/ftp/zvm61
#
# Uncomment this to allow local users to log in.
# local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
# write_enable=YES
...
```

- Set the vsftpd service to start at boot time with the **chkconfig** command and for this session with the **service** command:

```
# chkconfig vsftpd on
# service vsftpd start
Starting vsftpd for vsftpd: [  OK  ]
```

An anonymous FTP server should now be running with the z/VM 6.1 directory in `/cpdvd` (relative to the anonymous FTP root directory).

6.4.3 Testing the anonymous FTP server

Test the setup by FTPing in as anonymous from another system. You should see the `cpdvd/` directory:

```
# ftp gpok223
Connected to gpok223.endicott.ibm.com.
220 (vsFTPd 2.2.2)
Name (gpok223:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||6252|).
150 Here comes the directory listing.
-rw-r--r--    1 0        0        45088210 Nov 11 19:06 CD813270.ZIP
```

```
dr-xr-xr-x    2 0        0           24576 Nov 11 19:23 cpdvd
226 Directory send OK.
ftp> quit
```

This shows that the anonymous FTP server is working. You should now be able to continue with a z/VM installation via FTP, starting in 4.1, “Installing z/VM from DVD or FTP server” on page 28.

7

Installing SLES 11 SP1 on the cloner

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

In this chapter you will install SLES 11 SP1 onto the user ID `S11S1CLN`, which is referred to as the *cloner*. This section describes how to configure the cloner, which will provide the following functions:

- A SLES 11 SP1 install server: a tree of the RPMs and other files needed for installation.
- Cloning: the ability to copy a Linux system from a logged off source user ID to a target user ID with identical minidisks.
- NFS server: to export the install files and possibly other data and directories.

To achieve this, perform the following steps:

7.1 Creating the user ID S11S1CLN

In this section you will define the user ID S11S1CLN. Think of this as defining a virtual PC with memory and small disks and a Network Interface Card (NIC), but no operating system (yet).

To accomplish this, perform the following steps:

1. Log on to MAINT
2. Edit the USER DIRECT file:

   ==> **x user direct c**

   In the USER DIRECT file you can group statements that will be common to many user definitions in a construct called a *profile*. This profile can then become part of the user definitions using the INCLUDE statement. You used the existing profile TCPCMSU when you defined the LNXMAINT user.

3. Create a new profile named LNXDFLT. This will contain the user directory statements that will be common to all Linux user IDs. To save typing, you can use the **""** prefix commands to duplicate the IBMDFLT profile that should be on lines 37-49:

```
""037 ************************************************************
00038 *
00039 PROFILE IBMDFLT
00040   SPOOL 000C 2540 READER *
00041   SPOOL 000D 2540 PUNCH A
00041   SPOOL 000D 2540 PUNCH A
00042   SPOOL 000E 1403 A
00043   CONSOLE 009 3215 T
00044   LINK MAINT 0190 0190 RR
00045   LINK MAINT 019D 019D RR
00046   LINK MAINT 019E 019E RR
00047   LINK MAINT 0402 0402 RR
00048   LINK MAINT 0401 0401 RR
""049 *
```

4. Edit the duplicated profile by deleting the two LINK MAINT 040x lines, and inserting the lines that are in bold text below:

```
PROFILE LNXDFLT
  IPL CMS                                  1
  MACHINE ESA 4                            2
  CPU 00 BASE                              3
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1     4
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK LNXMAINT 192 191 RR                 5
  LINK TCPMAINT 592 592 RR                 6
```

Notes:

1 CMS will be IPLed when the user ID is logged onto.
2 Machine of type ESA with a maximum of four processors that can be defined.

3 Defines the base processor (**Note**: some workloads will benefit from a second virtual processor by adding another line with `CPU 01`).

4 Defines a virtual NIC connected to the VSWITCH named `VSW1` starting at virtual address `600`.

5 Provides read access to the LNXMAINT 192 disk as the user's 191 disk.

6 Provides read access to the TCPMAINT 592 disk, so that the user has access to TCPIP services such as an FTP client.

5. Go to the bottom of the file and add the definition for a new user ID named `S11S1CLN`. This user ID is given class B privilege, in order to run the **`FLASHCOPY`** command, class D privilege to run the **`QUERY ALLOC MAP`** command, and class E privilege to run the **`QUERY NSS`** command. Be sure to replace the volume labels shown in this example with the labels of your DASD:

```
*
USER S11S1CLN  LNX4VM 512M 1G BDEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3338 UM6290 MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 0321 3018 UM6289 MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 0001 3338 UM6293 MR LNX4VM LNX4VM LNX4VM
 MDISK 103 3390 0001 3338 UM6294 MR LNX4VM LNX4VM LNX4VM
 MDISK 104 3390 0001 3338 UM63A2 MR LNX4VM LNX4VM LNX4VM
 MDISK 105 3390 3339 3338 UM63A2 MR LNX4VM LNX4VM LNX4VM
```

This Linux user ID will have the following minidisks:

`100` A single 3390-3 for the root file system

`101-105` A mix of 3390-3s and 3390-9s that will be used to create a logical volume mounted over `/nfs/`. This stores the SLES 11 SP1 installation tree and the files associated with this book. Note that the 3390 at device address 6289 starts at cylinder 321 because the first 320 cylinders were used for the LNXMAINT 191 and 192 disks. Also note that minidisks 104 and 105 were allocated in 3338 cylinder chunks from the same 3390-9. This will work in the event that you have only 3390-3s.

6. Go back to the top of the file and search for string `USER $ALLOC$`. Add cylinder 0 of each of the four new volumes to this dummy user ID so they do not show up as gaps:

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 610RES R
 MDISK A02 3390 000 001 UV6283 R
 MDISK A03 3390 000 001 UV6284 R
 MDISK A04 3390 000 001 UM6289 R
 MDISK A05 3390 000 001 UM6290 R
 MDISK A06 3390 000 001 UM6293 R
 MDISK A07 3390 000 001 UM6294 R
 MDISK A08 3390 000 001 UM63A2 R
...
====> file
```

7. Run **`DISKMAP`** to check for overlaps and gaps. You should only see the two gaps that are in the default `USER DIRECT` file. These are expected. Press F3 to quit.

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
```

```
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  328 line(s) not displayed --------------------
====> F3
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 44 disk pages
```

You have now defined the user ID that will become the cloner.

7.2 Updating AUTOLOG1's PROFILE EXEC

The new Linux user ID, S11S1CLN, will need access to the VSWITCH. A **SET VSWITCH** command with the **GRANT** parameter can be added to AUTOLOG1's PROFILE EXEC to do this. Also, an **XAUTOLOG** statement can be added so the cloner 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 S11S1CLN to the section that grants access to the VSWITCH. Also add S11S1CLN to the section that **XATOLOG** the Linux user IDs:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
==> x profile exec f
/*************************/
/*  Autolog1 Profile Exec  */
/*************************/
'cp xautolog tcpip'                 /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'        /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant s11s1cln'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog s11s1cln'
'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 s11s1cln
Command complete
```

```
==> q vswitch acc
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-01-00-00-01
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
    Authorized userids:
      SYSTEM   S11S1CLN
  RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
  RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1  BACKUP
```

The new user ID, S11S1CLN, has now been granted access to the VSWITCH VSW1 for this session and at IPL time, and it automatically logged on at IPL time.

7.3 Preparing SLES 11 SP1 bootstrap files

To IPL a SLES 11 SP1 installation system, three bootstrap files must be copied and "punched" to the reader. Then you can IPL an install system from the reader (virtual address 00C). The three files are a kernel, a parameter file, and an initial RAMdisk. Think of these files as a PC Linux boot floppy or CD. Also, a small REXX EXEC is commonly used to clean out the reader, punch the three files, and IPL the reader.

1. Start an SSH session as root on the NFS server.
2. Change the directory to the mounted DVD. You should see a directory boot/ where the kernel and RAMdisk are located:

```
# cd /nfs/sles11sp1/dvd1
# ls -F
ARCHIVES.gz   README          gpg-pubkey-0dfb3188-41ed929b.asc  ls-lR.gz
COPYING       boot/           gpg-pubkey-307e3d54-481f30aa.asc  media.1/
COPYING.de    content         gpg-pubkey-3d25d3d9-36e12d04.asc  pubring.gpg
COPYRIGHT     content.asc     gpg-pubkey-7e2e3b05-4816488f.asc  suse/
COPYRIGHT.de  content.key     gpg-pubkey-9c800aca-481f343a.asc  suse.ins
ChangeLog     control.xml     gpg-pubkey-a1912208-446a0899.asc
INDEX.gz      directory.yast  gpg-pubkey-b37b98a9-486b702f.asc
NEWS          docu/           license.tar.gz
```

3. Change directory to boot/s390x/ and invoke the **ls** command. You should see the initial RAMdisk and kernel named initrd and vmrdr.ikr.

```
# cd boot/s390x
# ls -l initrd vmrdr.ikr
-r--r--r-- 1 root root 13323355 May 20 06:59 initrd
-r--r--r-- 1 root root  6885632 May 20 06:59 vmrdr.ikr
```

4. FTP to z/VM and log in as LNXMAINT. Copy the SLES 11 SP1 kernel (the file vmrdr.ikr copied as SLES11S1 KERNEL) and the initial RAMdisk (the file initrd copied as SLES11S1 INITRD). These files must have a record format of fixed 80-byte records and be transferred in binary. This format can be set with the **site fix 80** FTP subcommand (if this subcommand fails, try **quote site fix 80**). Following is an example:

```
# ftp 9.60.18.249
Connected to 9.60.18.249.
220-FTPSERVE IBM VM Level 610 at GPOK249.ENDICOTT.IBM.COM, 08:59:44 EST
THURSDAY 2009-11-26
```

```
220 Connection will close if idle for more than 5 minutes.
Name (9.60.18.249:root): lnxmaint
331 Send password please.
Password: lnx4vm
230 LNXMAINT logged in; working directory = LNXMAINT 191
Remote system type is z/VM.
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> bin
200 Representation type is IMAGE.
ftp> site fix 80
200 Site command was accepted.
ftp> put vmrdr.ikr SLES11S1.KERNEL
local: vmrdr.ikr remote: SLES11.KERNEL
...
ftp> put initrd SLES11S1.INITRD
local: initrd remote: SLES11.INITRD
...
ftp> quit
```

5. Go back to your 3270 session. Log off `MAINT` if you are still logged on and then log on to `LNXMAINT`.
6. Besides the kernel and RAMdisk that you just copied, the file `SLES11S1 EXEC` should exist on the `LNXMAINT 192` disk. Use the **`FILELIST`** command to verify that the kernel and RAMdisk were copied in fixed 80-byte record format. You should see the following files:

```
==> filel sles11s1 * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=18 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SLES11S1 EXEC     D1 V         72         10          1  8/31/10 10:59:48
      SLES11S1 INITRD   D1 F         80     166542       3253  8/31/10 10:57:37
      SLES11S1 KERNEL   D1 F         80      86071       1390  8/31/10 10:57:28
```

7. Quit by pressing F3.
8. Verify that the file `SLES11 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:

```
==> type sles11s1 exec d

/* EXEC to punch SLES-11 SP1 install system to reader and IPL from it */
Address 'COMMAND'
'CP SPOOL PUN *'
'CP CLOSE RDR'
'CP PURGE RDR ALL'
'PUNCH SLES11S1   KERNEL * (NOHEADER'
'PUNCH' Userid() 'PARM-S11 * (NOHEADER'
'PUNCH SLES11S1   INITRD * (NOHEADER'
'CP CHANGE RDR ALL KEEP'
'CP IPL 00C CLEAR'
```

9. A sample parameter file named `SAMPLE PARM-S11` is provided to save typing. These are the parameters, especially networking values, that will be used to install SLES.
10. View the SLES 11 SP1 sample parameter file using the **`TYPE`** command:

```
==> type sample parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
```

```
HostIP=n.n.n.n Hostname=yourhost.example..com
Gateway=n.n.n.n Netmask=255.255.255.0
Broadcast=n.n.n.n Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=n.n.n.n
portname=whatever
portno=0
Install=nfs://n.n.n.n/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

11. Copy the sample parameter file to a new file with the file name `S11S1CLN`. Edit the file and update the networking variables with the correct values for your site. Refer to the worksheet in 2.7.3, “Linux resources worksheet” on page 18. The fields you should change are in ***bold-italics***. The examples used in this book are as follows:

```
==> copy sample parm-s11 d s11s1cln = =
==> x s11s1cln parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.60.18.223 Hostname=gpok223.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.240/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

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

> **Note:** SLES 11 SP1 documentation states that the VNC password should be six to eight characters. However, actual installation process requires it to be at least eight characters long. If it is shorter, the installer will prompt for the password.

Now you are ready to start the cloner installation.

7.4 Installing the cloner

In this section you will install the cloner Linux image onto the new user ID `S11S1CLN`.

1. Log onto `S11S1CLN`, the cloner user ID. You should see a virtual NIC being created at virtual address `600`, and the common PROFILE EXEC running that creates two VDISKS for swap spaces at virtual addresses 300 and 301:

```
00: NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0003 RDR,   NO PRT,   NO PUN
LOGON AT 14:55:00 EST THURSDAY 11/26/09
z/VM V6.1.0    2009-11-19 13:47

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
```

```
Do you want to IPL Linux from minidisk 100? y/n
```

2. Answer no (**n**) to the question asking you to IPL Linux from 100:

```
Do you want to IPL Linux from minidisk 100? y/n
n
```

3. You may want to look at the PROFILE EXEC using the **TYPE** command:

```
==> type profile exec
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 300 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else  /* user is interactive -> prompt */
 do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
 end /* else */
```

4. Run the SLES11S1 EXEC to purge the reader, punch the bootstrap files, and IPL from the reader. You should see the Linux RAMdisk getting loaded into memory. Look for the contents of the parameter file you created:

```
==> sles11s1
NO FILES PURGED
RDR FILE 0001 SENT FROM S11S1CLN PUN WAS 0001 RECS 086K CPY  001 A NOHOLD
NOKEEP
RDR FILE 0002 SENT FROM S11S1CLN PUN WAS 0002 RECS 0009 CPY  001 A NOHOLD
NOKEEP
RDR FILE 0003 SENT FROM S11S1CLN PUN WAS 0003 RECS 167K CPY  001 A NOHOLD
NOKEEP
0000003 FILES CHANGED
0000003 FILES CHANGED
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32.12-0.7-default (geeko@buildhost) (gcc version 4.3.4
Ýgcc-4_
3-branch revision 152973¨ (SUSE Linux) ) #1 SMP 2010-05-20 11:14:20 +0200
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
...
pcpu-alloc: Ý0¨ 56 Ý0¨ 57 Ý0¨ 58 Ý0¨ 59 Ý0¨ 60 Ý0¨ 61 Ý0¨ 62 Ý0¨ 63
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 129280
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
TERM=dum
b                    HostIP=9.60.18.223 Hostname=gpok223.endicott.ibm.com
                     Gateway=9.60.18.129 Netmask=255.255.255.128
                     Broadcast=9.60.18.255 Layer2=0
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601
DataChannel=0.
```

```
0.0602              Nameserver=9.0.2.11 portname=whatever portno=0

Install=nfs://9.60.18.240/nfs/sles11sp1/SLES-11-SP1-DVD-s39
0x-GM-DVD1.iso      UseVNC=1  VNCPassword=12345678
                    InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
PID hash table entries: 2048 (order: 2, 16384 bytes)
...
```

Important: If you see the following output on your 3270 window, there is a problem:

```
Activating manual setup program.

>>> Linuxrc v3.3.34 (Kernel 2.6.27.19-5-default) <<<

Main Menu

1) Start Installation
2) Settings
3) Expert
4) Exit or Reboot

>
```

Often, this is because the install program cannot successfully mount the NFS file system, or it can mount it, but cannot find the SLES 11 SP1 install files. Also, this can result when there is not enough memory (though the example uses 512 MB, which should be plenty).

If you get these messages, you may want to immediately restart the install process with the **IPL 00C** command, which IPLs from the reader. Watch the console messages carefully.

You may choose to enter **3** (Expert) then **7** (Start shell). Then you should be able to issue commands to see if the network is up, and so on.

5. A VNC server process will be started. You should see the messages:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.60.18.223, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.60.18.223:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***
```

6. From your workstation you can open a Java™-enabled browser to access YaST2 at the specified URL. The logon prompt in Figure 7-1 on page 114 shows VNC access through a Java-enabled browser.

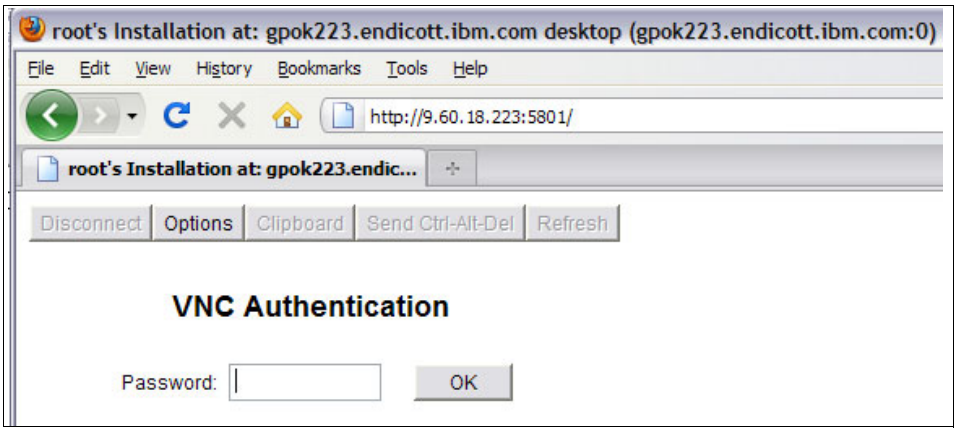


Figure 7-1 VNC viewer through a Java-enabled browser

In addition to a browser, you can also use a standalone VNC viewer if desired (RealVNC in this example). The server to connect to would be 9.60.18.223:1 in this example, as shown on the left side of Figure 7-2. Enter the password specified in the parameter file (12345678 in this example).

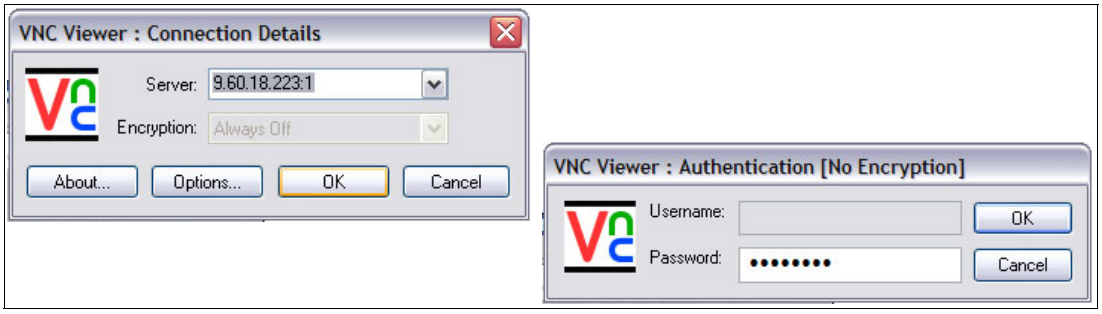


Figure 7-2 Using the VNC viewer

7. You should see a window entitled Welcome. Select your Language and Keyboard. After reading the Licence Agreement, click the check-box **I Agree to the License Terms**, then click **Next**.
8. At the Disk Activation window, click **Configure DASD Disks**.
9. At the DASD Disk Management window:
   a. Click **Select** or **Deselect** for all seven read-write minidisks (100-104, 300 and 301).
   b. Click **Activate** on the Perform Action pop-up menu. The DASD will be activated quickly, as shown on Figure 7-3 on page 115.
   c. Click **Select** or **Deselect** to deselect minidisks 300 and 301 so that six minidisks, 100-105, are selected.
   d. Click **Format** on the Perform Action pop-up menu.
   e. Click **OK** to the query to format six disks in parallel. Then click **Yes** to confirm. The DASD will be formatted in parallel. This will take a number of minutes.
   f. Click **Next** when the formatting is complete.
   g. In the Disk Activation window, click **Next**.

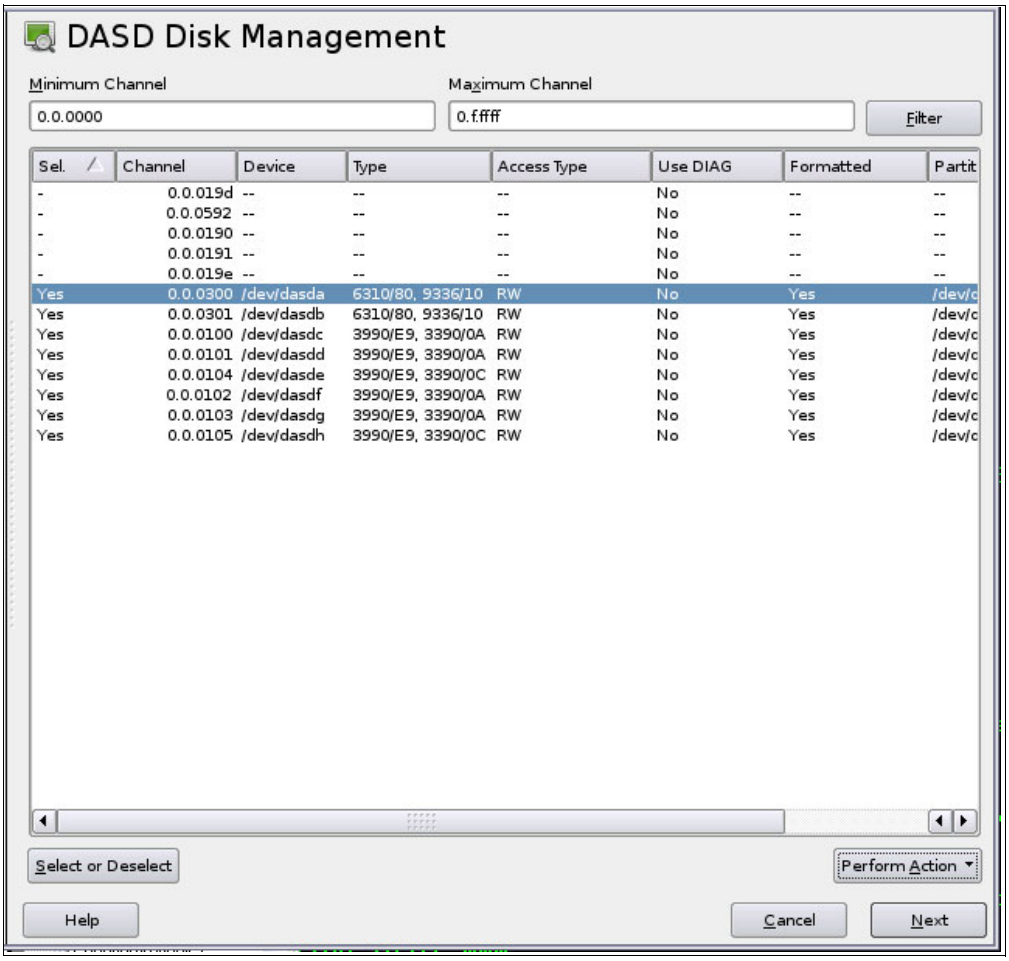


Figure 7-3 Activating DASD on the cloner

10. In the Installation Mode window, accept the default of **New installation** and click **Next**.
11. In the Clock and Time Zone window, select your time settings and click **Next**.
12. In the Installation Settings window, click **Partitioning** and the Preparing Hard Disk window will appear.
13. Accept the default of **Customer Partitioning (for experts)** and click **Next**. The Expert Paritioner window will appear.
14. In the System View column on the left, click a plus sign next to **Hard disks** to expand a list of all available minidisks, as shown in Figure 7-4 on page 116.

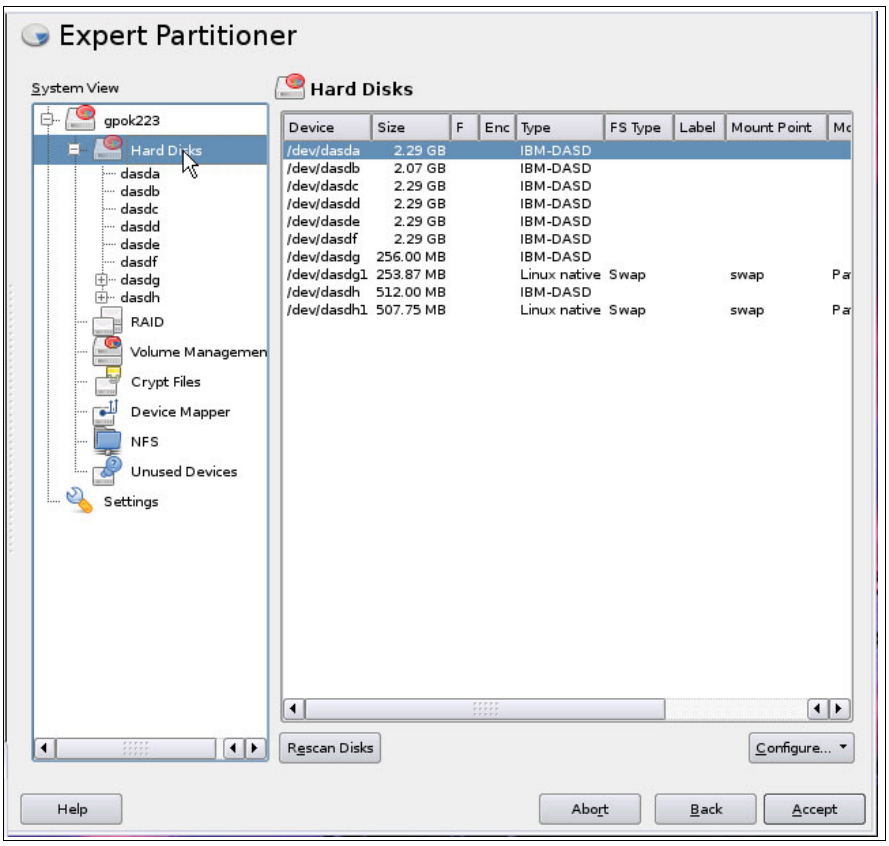


Figure 7-4 Expert partitioner - hard disks

15. Select **dasda** in the System View column and click **Add** to add a partition, as shown in Figure 7-5 on page 117. The window Add Partition on /dev/dasda will appear.

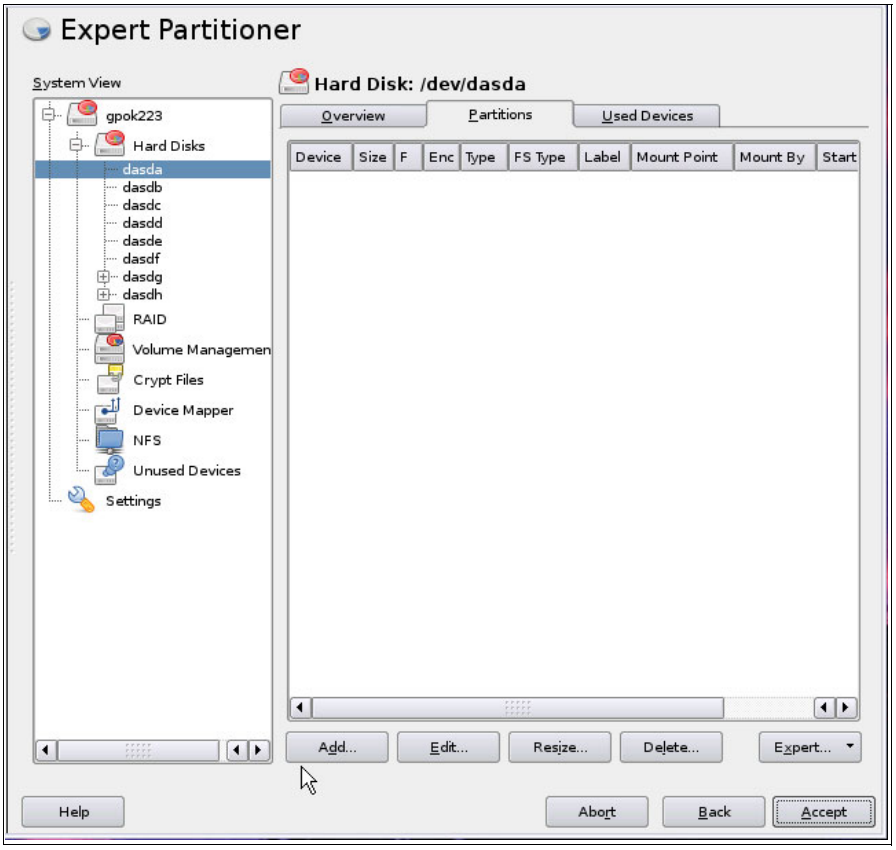


Figure 7-5 Expert Partitioner - Add Partition

16. Set partition size to Maximum Size and click **Next**.
17. On the next window, accept the defaults (Format partition, Ext3 file system, Mount partition and / (the root file system)) as the Mount Point. See Figure 7-6 on page 118. Click **Finish**.

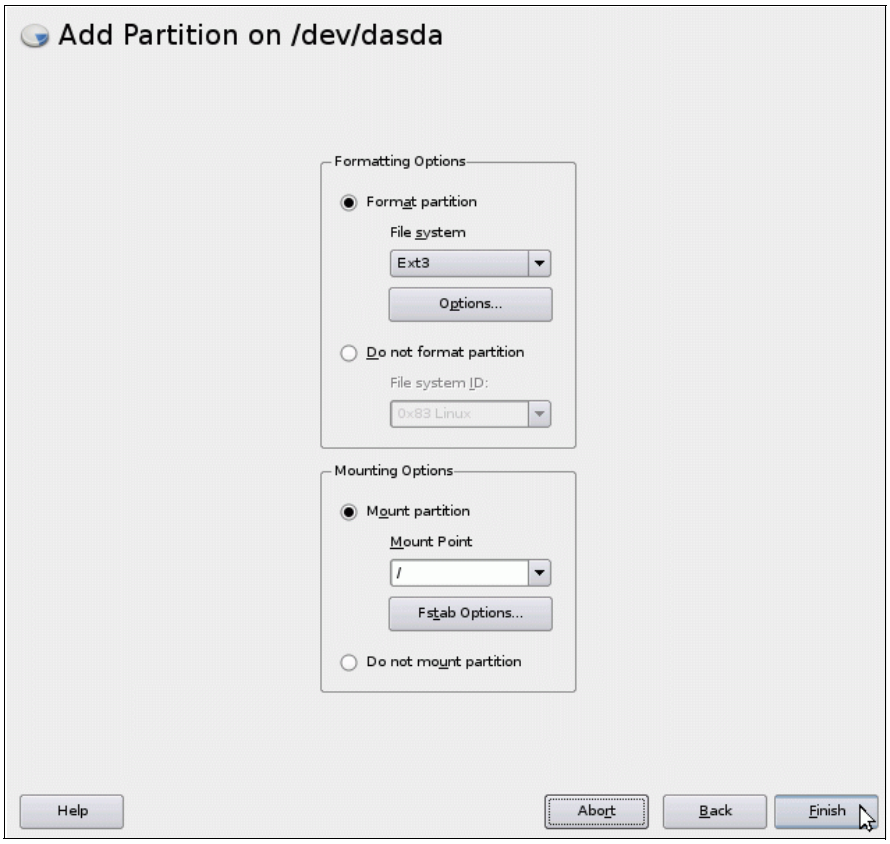


Figure 7-6 Add partition on /dev/dasda

18. Back in the Expert Partitioner panel, select **dasdb** in the left column and click **Add** to add a partition to it.
19. On the Add Partition on /dev/dasdb window, select **Maximum Size** for the partition and click **Next**.
20. The disk `/dev/dasdb1` will be used in a logical volume. Select **Do not format partition** in the Formatting Options area and **Do not mount partition** in the Mounting Options area, as shown in Figure 7-7 on page 119. Click **Finish**.

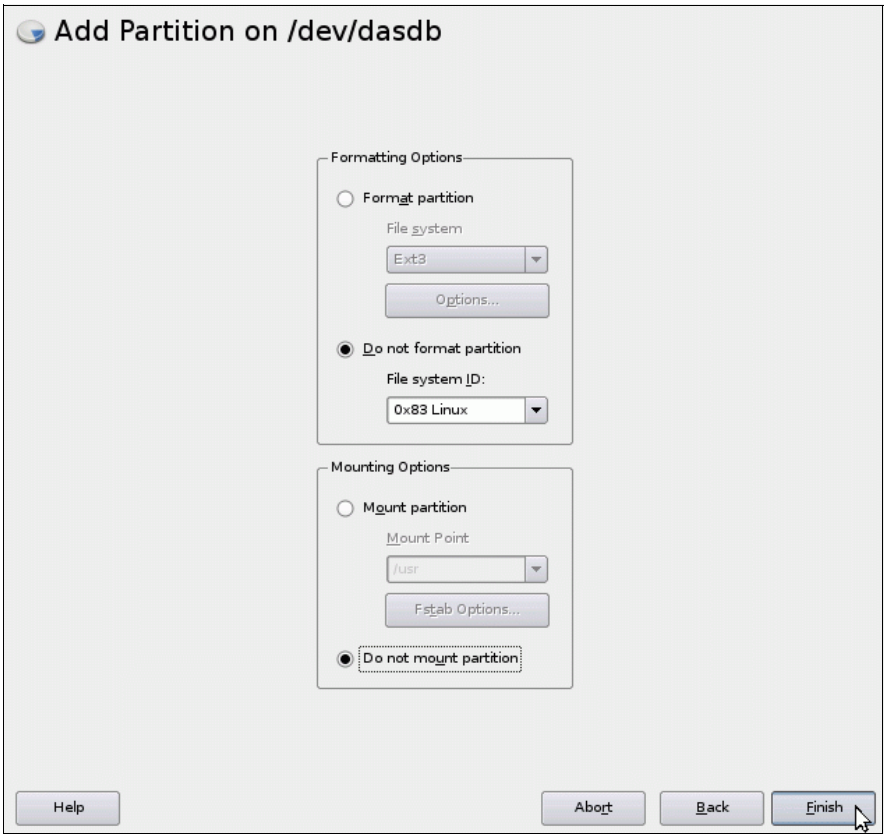


Figure 7-7 Partitioning /dev/dasdd to mount over /backup/

21. You should now see a `/dev/dasdb1` in the Expert Partitioner window. Repeat the previous three steps for `/dev/dasdc`, `/dev/dasdd`, `/dev/dasde` and `/dev/dasdf`. This will give you five partitions `/dev/dasdb1-dasdf1` for use with LVM.

22. Select **Volume Management** in the System View column. There should be no volume groups defined, as shown in “Volume Management” on page 120. Click **Add Volume Group**. The Add Volume Group window will appear.

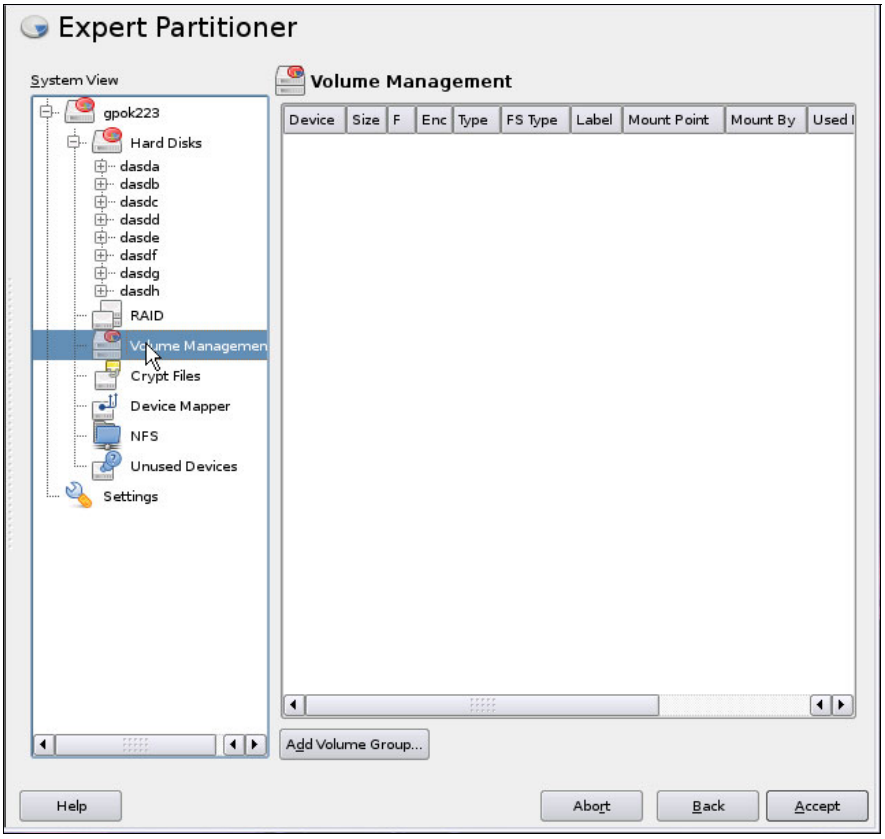


Figure 7-8 Volume Management

23. In the Add Volume Group window, set the Volume Group Name to `nfs-vg`. Accept the default Physical Extent Size of 4 MB. Click **Add All** to add `dasdb1-dasdf1` to a newly defined volume group. Click **Finish.**
24. Back in the Expert Partitioner window, click the plus sign next to Volume Management in the System View. You should see the new volume group, `nfs-vg`.
25. Click it, and the Logical Volumes tab will appear. There should be no logical volumes defined. Click **Add** to add a logical volume.
26. In the “Add Logical Volume on /dev/nfs-vg” window, enter `nfs-lv` as a logical volume name and click **Next**.
27. Accept the defaults of Maximum Size (11.23 GB) and Stripes of 1. Click **Next**.
28. In the “Add Logical Volume on /dev/nfs-vg” window, choose **Format Partition** and select a File system type of `ext2` (a file system type of ext2 is recommended on the assumption that this file system will rarely be written to. You may choose to use ext3). Specify a Mount Point of `/nfs`. Click **Finish**.
29. Back in the Expert Partitioner window, click the plus sign next to `nfs-vg` in the System View column, Then click the **nfs-lv** that you just defined. A summary of the logical volume will be shown, as in Figure 7-9 on page 121.
30. Click **Accept** to close the partitioner.

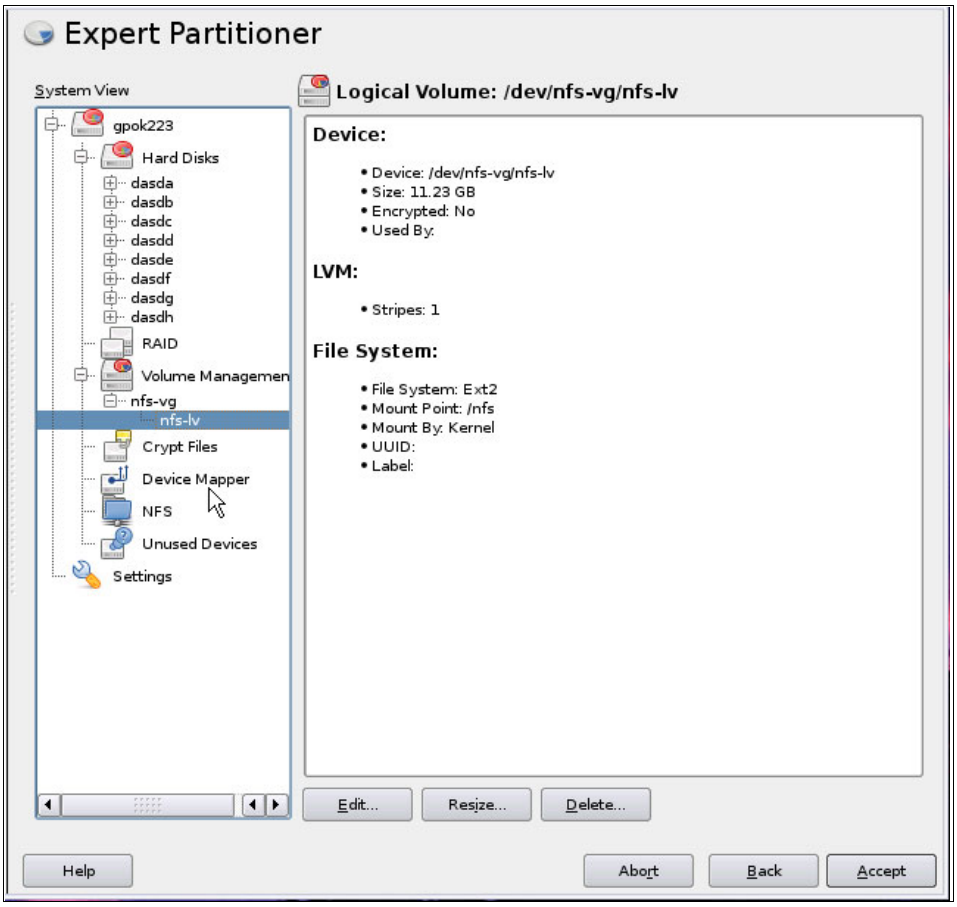


Figure 7-9 Expert Partitioner - Logical Volume

31. Back on the Installation Settings window, click **Software**. You may get a warning window about low disk space. This is OK, it will be fixed. Deselect all items except **Base Server, 32bit Runtime Environment, Help and Support Documentation**, and **Minimal System** as shown in Figure 7-10 on page 122. To deselect a box, click it once. There is always a short delay between selecting and deselecting a group and refreshing disk usage window on the bottom right. Click **Ok**.

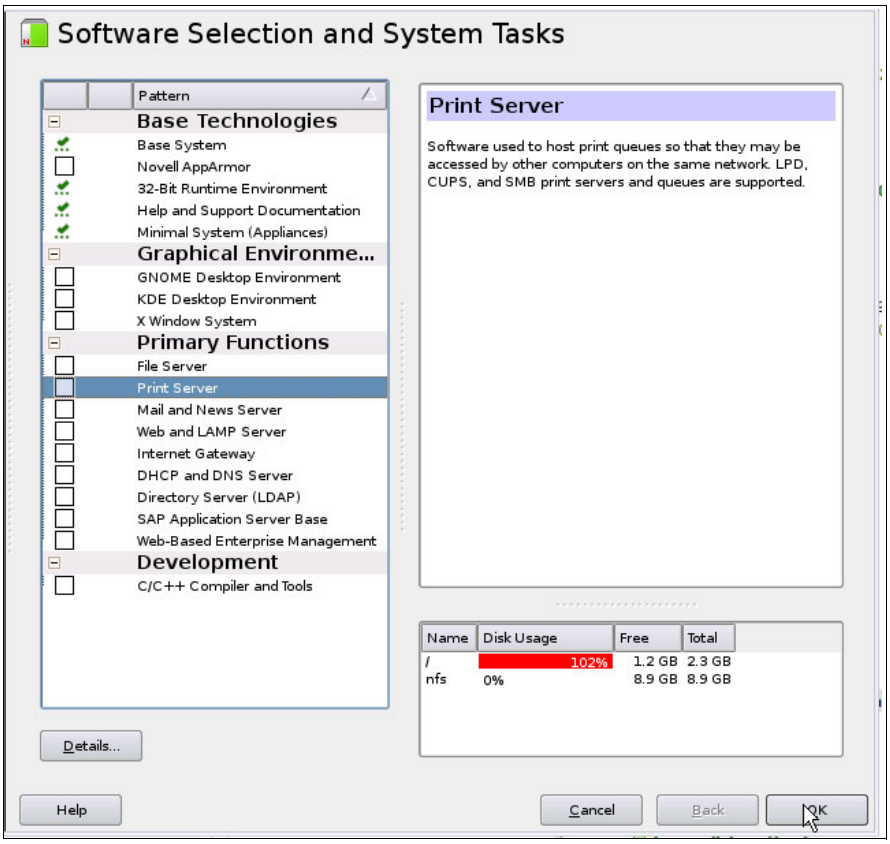


Figure 7-10 Choosing software groups

32. In the Expert tab, click **Default Runlevel** near the bottom. Choose **3: Full multiuser with network** as shown in Figure 7-11. Click **OK**. If you receive a VNC warning, click **Yes**.

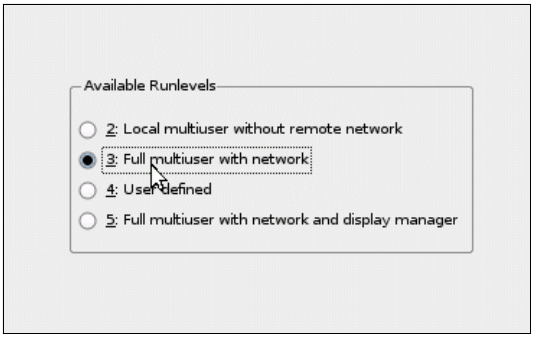


Figure 7-11 Setting default run level

33. In the Installation Settings window, click **Install**.

34. On the Confirm Installation window, click **Install**. This will begin the process of laying down RPMs onto your system. Copying the RPMs should take about 5-20 minutes, depending on network and disk speeds. When copying of the RPMs is done, a few more windows will pass by and then your VNC viewer session will close.

Completing the cloner installation

1. Go back to your 3270 session. You will see messages indicating that the Linux image is being restarted. You may need to clear the window a number of times. At the end of the re-IPL, the VNC server is started again.

```
...
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.60.18.223, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.60.18.223:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***
```

2. Start the VNC viewer session again.
3. On the Password for the System Administrator “root” window, set the root password twice and click **Next**. You may get a warning about the strength of the password.
4. On the Hostname and Domain Name window, the Hostname and Domain Name fields should be filled in by values from the parameter file (`S11S1CLN PARM-S11`). Uncheck the box Change Hostname via DHCP. Click **Next**.
5. On the Network Configuration window, in the Firewall section, click **disable** to disable the firewall. Click **Next**.
6. On the Test Internet Connection window, if you do not have an Internet connection, select **No, Skip This Test**. However, note that Novell recommends you perform this test if possible. Click **Next**.
7. On the Network Services Configuration window, accept the defaults and click **Next**. A certificate will be created.
8. On the User Authentication Method window, select **Local (/etc/passwd)** and click **Next**.
9. On the “Add a new local user” window, add a non-root userid for the primary system administrator(s) for this system and click **Next**.
10. At the Release Notes window, we recommend that you read the information. Click **Next**.
11. At the Hardware Configuration window, click **Next**.
12. At the Installation Complete window, deselect the check box and click **Clone this system for Autoyast**, then click **Finish**. The VNC viewer session will end.
13. Go back to the 3270 session and you may have to clear the window a few times. When you see the login prompt, **`DISCONNECT`** using the **`#CP`** prefix:

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

You have now installed the cloner. You should be able to access the new system using SSH.

7.4.1 Verifying the installation

Using an SSH client, such as PuTTY, start a new SSH session to the cloner as **root**. Verify some settings using the **`df -h`** and **`swapon -s`** commands. You should see the following file systems mounted and the two active swap spaces:

```
# df -h
```

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  1.3G  892M  60% /
devtmpfs              248M  152K  247M   1% /dev
tmpfs                 248M     0  248M   0% /dev/shm
/dev/mapper/nfs--vg-nfs--lv
                       12G   29M   11G   1% /nfs
# swapon -s
Filename                                Type            Size    Used
Priority
/dev/dasdd1                             partition       259956  0       -1
/dev/dasde1                             partition       519924  0       -2
```

Linux is now installed on the cloner. The next step is to configure it.

7.5 Configuring the cloner

Now that your cloner is installed, it must be configured.

7.5.1 Copying files to the cloner

The cloner can now be configured to replace the NFS (PC) server to make the SLES 11 SP1 install files and the files associated with this book available with NFS. Copy the following files from the NFS server to the cloner:

- The SLES 11 SP1 DVD ISO image(s) - the first ISO image is needed, the second is optional. But you may want to copy it now in case you need it later.
- The tar file associated with this book - `virt-cookbook-S11.tgz`

Perform the following steps:

1. Create the directory `/nfs/sles11sp1/`:

   ```
   # cd /nfs
   # mkdir sles11sp1
   ```

2. Change the directory into the `sles11sp1/` directory and copy the DVD ISO image(s) from the PC NFS server to this directory. This step may take some time.

   ```
   # cd sles11sp1
   # scp 9.60.18.240:/nfs/sles11sp1/*.iso .
   The authenticity of host '9.60.18.240 (9.60.18.240)' can't be established.
   RSA key fingerprint is f5:43:ce:f3:44:35:81:b2:f5:9a:5e:06:f6:fb:46:56.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added '9.60.18.240' (RSA) to the list of known hosts.
   Password:
   SLES-11-SP1-DVD-s390x-GM-DVD1.iso               2%   67MB  11.2MB/s   04:07 ETA
   ```

3. Change the directory to `/nfs/` and copy the files associated with this book from the NFS server to this directory:

   ```
   # cd /nfs
   # scp 9.60.18.240:/nfs/SG247931.tgz .
   Password:
   SG247931.tgz                                  100%   17KB  16.6KB/s   00:00
   ```

4. Untar the file `SG247931.tgz`:

```
# tar xzvf SG247931.tgz
virt-cookbook-S11SP1/
virt-cookbook-S11SP1/vm/
virt-cookbook-S11SP1/vm/cpformat.exec
virt-cookbook-S11SP1/vm/profile.exec
virt-cookbook-S11SP1/vm/sample.parm-s11
virt-cookbook-S11SP1/vm/sles11s1.exec
virt-cookbook-S11SP1/vm/swapgen.exec
virt-cookbook-S11SP1/vm/chpw610.xedit
virt-cookbook-S11SP1/README.txt
virt-cookbook-S11SP1/clone.sh
```

5. Change into the newly-created directory `virt-cookbook-S11/` and view the files:

```
# cd virt-cookbook-S11SP1/
# ls -F
README.txt  clone.sh*  vm/
```

6. Copy the `clone.sh` file to `/usr/local/sbin/`. It will be used later for cloning:

```
# cp clone.sh /usr/local/sbin
```

You have now copied the files you will need from the PC NFS server.

7.5.2 Resetting the install source location

When SLES 11 SP1 is installed, the location of the install *source* is remembered. In this case, it is the PC NFS server. Now that the DVD ISO image has been copied from the PC NFS server to the cloner, you can reset the install source location to point to `/nfs/sles11sp1` in the local file system. To do this task, perform the following steps:

1. Invoke yast:

```
# yast
```

2. Accept the default of **Software** in the left column and using the arrow keys, select **Software Repositories** in the right column and press Enter.

```
  +------------------------------------------------------------------------+
  ¦                          YaST2 Control Center                          ¦
  +------------------------------------------------------------------------+

  +--------------------+ +-------------------------------------------------+
  ¦Software            ¦ ¦Online Update                                    ¦
  ¦Hardware            ¦ ¦Software Management                              ¦
  ¦System              ¦ ¦Add-On Products                                  ¦
  ¦Network Devices     ¦ ¦Installation into Directory                      ¦
  ¦Network Services    ¦ ¦Media Check                                      ¦
  ¦Security and Users  ¦ ¦Online Update Configuration                      ¦
  ¦Support             ¦ ¦Patch CD Update                                  ¦
  ¦Miscellaneous       ¦ ¦Software Repositories                            ¦
...
```

3. At the top of the Configured Software Repositories panel, you should see a single entry for SLES 11 SP1. Add a new installation source by using the Tab key to move to the Add button at the bottom and press Enter.

```
Configured Software Repositories
  +--------------------++-------------------------------------------------+
```

```
|---Services          ||Priority     |Enabled|Autorefresh|Name            |
|---Repositories      || 99 (Default)|   x   |     x     |SUSE-Linux-Enter|
|                     ||                                                    |
|                     ||                                                    |
|                     ||                                                    |
|                     ||                                                    |
|                     |++--------------------+--------------------------+
|                     |+-------------------------------------------------+
|                     || SUSE-Linux-Enterprise-Server-11-SP1 11.1.1-1.152 |
|                     || URL: iso:///?iso=SLES-11-SP1-DVD-s390x-GM-DVD1.is|
|                     || Category: YaST                                   |
|                     ||                                                  |
|                     || Properties                                       |
|                     || [x] Enabled              Priority                |
|                     || [x] Automatically Refresh   v 99^                |
|                     |+-------------------------------------------------+
|                     |[Add][Edit][Delete]           [GPG Keys...][Refreshâ]
+---------------------+
```

4. Use the Tab key to move to Local ISO Image and press Enter to select. Move to the Next field and press Enter.

```
( ) Scan Using SLP...
( ) Specify URL...
( ) FTP...
( ) HTTP...
( ) HTTPS...
( ) SMB/CIFS
( ) NFS...
( ) CD...
( ) DVD...
( ) Hard Disk...
( ) USB Mass Storage (USB Stick, Disk)...
( ) Local Directory...
(x) Local ISO Image...
```

5. Use the Tab key to move to Local ISO Image and select it by pressing Enter (or the space bar).
6. Use the Tab key to move to Next and press Enter.
7. Set the Path to ISO Image to `/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso`. Tab to Next and press Enter:

```
Repository Name
  ------------------------------------------------------------------------
  Path to ISO Image
  /nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso----------------[Browse...]
```

8. In the License Agreement panel, move to Yes and press Enter to select. Move to Next and press Enter:

```
(x) Yes, I Agree to the License Agreement
( ) No, I Do Not Agree
```

9. You should now see the Configured Software Repositories panel again, this time with two entries. The new repository will be the second entry. The old one (on the PC NFS server) will be the first. Select the first one and use the Tab key to move to Delete. Press Enter.
10. Confirm the delete by choosing **Yes**.
11. If the new repository is not enabled, use the Tab key to move to Enabled and select it.

```
+-----------------------------------------------+
¦ SUSE-Linux-Enterprise-Server-11-SP1 11.1.1-1.152 ¦
¦ URL: iso:///?iso=SLES-11-SP1-DVD-s390x-GM-DVD1.is¦
¦ Category: YaST                                ¦
¦                                               ¦
¦ Properties                                    ¦
¦ [x] Enabled              Priority             ¦
¦ [ ] Automatically Refresh   v 99^             ¦
+-----------------------------------------------+
```

12. Finish with the Software Repositories dialog by pressing the F10 key.
13. Leave YaST by moving to **Quit**.

You have now changed the cloner to point to the local file system, /nfs/sles11sp1/, for the SLES 11 SP1 software repository.

7.5.3 Configuring the NFS server

To configure the NFS server to make the SLES 11 SP1 install directory available using NFS, perform the following steps:

1. The NFS server will be used to export the SLES11 ISO image. The service needed is named nfsserver. Install it with the **zypper install** command:

```
# zypper install nfs-kernel-server
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  nfs-kernel-server

1 new package to install.
Overall download size: 96.0 KiB. After the operation, additional 248.0 KiB will
be used.
Continue? [y/n/?] (y): y
Retrieving package nfs-kernel-server-1.2.1-2.6.6.s390x (1/1), 96.0 KiB (248.0
KiB unpacked)
Installing: nfs-kernel-server-1.2.1-2.6.6 [done]
```

2. Add two lines at the bottom of the /etc/exports file to allow the directories /nfs/virt-cookbook-S11SP1/ and /nfs/sles11sp1/ to be mounted:

```
# cd /etc
# vi exports
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.
/nfs/virt-cookbook-S11SP1       *(ro,sync,no_subtree_check)
/nfs/sles11sp1                  *(ro,sync,no_subtree_check)
```

   You may want to look at the /etc/exports man page with the command **man exports** to understand the hosts and options specified.

3. Turn on the service using the **chkconfig** command:

```
# chkconfig nfsserver on
```

When you restart the cloner later, the NFS server will be started.

4. Start the NFS server for this session:

```
# service nfsserver start
Starting kernel based NFS server: idmapd mountd statd nfsd sm-notify
done
```

You can retire the PC NFS server at this point because all necessary files are on the S11S1CLN NFS server.

7.5.4 Turning off unneeded services

There are a number of services which are started in an SLES11 minimum system. They can be viewed using the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog             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
haldaemon               0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer            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
network-remotefs        0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver               0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
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
rpcbind                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
smartd                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
splash                  0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early            0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
xinetd                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

In order to keep the cloner as lean as possible in terms of processor usage, some of these services can be turned off. Turn off the following services using the **chkconfig** command:

```
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
# chkconfig xinetd off
```

You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog             0:off  1:off  2:on   3:on   4:off  5:on   6:off
haldaemon               0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer            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
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver               0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                 0:off  1:off  2:off  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
```

7.5.5 Applying service if necessary; Online Update

You may want to apply service using YaST Online Update. Internet access was not available during the writing of this book, so step-by-step details are not available.

If you have access to the Internet, or an online update source, invoke **yast** → **Software** → **Online update**.

7.5.6 Installing the cmsfs package

The clone.sh script requires the cmsfs package, written by Rick Troth, in order to read CMS files. This package is part of SLES11 distribution. Install it using the **zypper install** command:

```
# zypper install cmsfs
...
Continue? [y/n/?] (y): y
Retrieving package cmsfs-1.1.8-6.2.s390x (1/1), 33.0 KiB (148.0 KiB unpacked)
Installing: cmsfs-1.1.8-6.2 [done]
```

You should see some windows flash by as the cmsfs RPM is installed:

```
# rpm -q cmsfs
cmsfs-1.1.8-6.2
```

To test that the cmsfs package is properly installed, see if you can read the S11S1CLN PARM-S11 parameter file. First you need to bring the 191 disk online using the **chccwdev -e** command. Then view the DASD that the system knows about using the **lsdasd** command:

```
# chccwdev -e 191
Setting device 0.0.0191 online
Done
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2122MB    543240
0.0.0102   active      dasdc     94:8    ECKD  4096   2347MB    600840
0.0.0300   active      dasdd     94:12   FBA   512    256MB     524288
0.0.0301   active      dasde     94:16   FBA   512    512MB     1048576
0.0.0103   active      dasdf     94:20   ECKD  4096   2347MB    600840
0.0.0104   active      dasdg     94:24   ECKD  4096   2347MB    600840
0.0.0105   active      dasdh     94:28   ECKD  4096   2347MB    600840
0.0.0191   active      dasdi     94:32   ECKD  4096   210MB     54000
```

Test some of the cmsfs utilities. The **cmsfslst** command lists files on a minidisk:

```
# cmsfslst -d /dev/dasdi
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
         DIRECTOR PO F         64         12          1  8/31/2010 12:41:05
         ALLOCMAP PO F       4096          2          2  8/31/2010 12:41:05
CHPW610  XEDIT    D1 V         72        190          3  8/31/2010  9:20:03
CPFORMAT EXEC     D1 V         79        252          3  8/31/2010  9:20:03
PROFILE  EXEC     D1 V         63         17          1  8/31/2010  9:20:03
PROFILE  XEDIT    D1 V         45         17          1  8/31/2010  9:20:03
SAMPLE   PARM-S11 D1 V         69         11          1  8/31/2010 12:41:05
SLES11S1 EXEC     D1 V         72         10          1  8/31/2010 11:59:48
SLES11S1 INITRD   B1 F         80     166542       3253  8/31/2010 11:57:37
SLES11S1 KERNEL   B1 F         80      86071       1390  8/31/2010 11:57:28
SWAPGEN  EXEC     D1 V         72        467          6  8/31/2010  9:20:03
S11S1CLN PARM-S11 D1 V         73          9          1  8/31/2010 12:40:26
```

The **cmsfscat** command types the contents of a file:

```
# cmsfscat -d /dev/dasdi -a s11s1cln.parm-s11
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.60.18.223 Hostname=gpok223.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.240/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

7.5.7 Enabling the vmcp and cmm modules

The **vmcp** module and command allows z/VM CP commands to be issued from Linux. It is critical to the functioning of the `clone.sh` script.

The cmm module allows Linux to do cooperative memory management (aka CMM1) with z/VM. This feature must be enabled on both the Linux and z/VM sides for it to function.

To configure the vmcp and cmm modules to be loaded at boot time, edit the file `/etc/sysconfig/kernel` and add the module names to the variable `MODULES_LOADED_ON_BOOT` (around line 30):

```
# cd /etc/sysconfig
# vi kernel   // add vmcp to MODULES_LOADED_ON_BOOT
...
## Type:               string
## ServiceRestart:     boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
# You will find a few default modules for hardware which
# can not be detected automatically
#
MODULES_LOADED_ON_BOOT="vmcp cmm"
...
```

Save the file and you should be able to issue CP commands using **vmcp** after your system is rebooted.

7.5.8 Setting the system to halt on SIGNAL SHUTDOWN

The Ctrl-Alt-Del key sequence is simulated by z/VM when it issues a **SIGNAL SHUTDOWN** command. Rather than rebooting, you want your system to halt (shut down). Change this setting by changing `shutdown -r` to `shutdown -h` in the `/etc/inittab` file:

```
# 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.

7.5.9 Modifying the zipl.conf file

Two changes are recommended to the default `zipl.conf` file:

1. Add the parameters `vmpoff=LOGOFF` and `vmhalt=LOGOFF`. These instruct the z/VM user ID to be logged off when Linux is shut down. This can be convenient for shutting the z/VM system down more efficiently and also for getting a refreshed 3270 emulator session.
2. The `timeout` value is modified to 3 seconds. When SLES 11 SP1 boots, the default time that is allowed to enter a menu command is 10 seconds. Because the default value is most commonly used, these 10 seconds delay the starting of Linux. It is backed up to three seconds.

Make a backup copy of the file and add the string `vmpoff=LOGOFF` to the parameter line in the `[Linux]` section and set `timeout=3` in the menu section:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf   // add the vmpoff and vmhalt, change timeout to 3
# Modified by YaST2. Last modification on Tue Aug 31 14:51:51 EDT 2010
[defaultboot]
defaultmenu = menu

###Don't change this comment - YaST2 identifier: Original name: linux###
[SLES11_SP1]
    image = /boot/image-2.6.32.12-0.7-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.32.12-0.7-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 vmpoff=LOGOFF
vmhalt=LOGOFF TERM=dumb"

###Don't change this comment - YaST2 identifier: Original name: failsafe###
[FailsafeV1]
    image = /boot/image-2.6.32.12-0.7-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.32.12-0.7-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb
x11failsafe"
```

```
:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 3
    1 = SLES11_SP1
    2 = FailsafeV1
    3 = ipl

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
    image = /boot/image
    target = /boot/zipl
    ramdisk = /boot/initrd,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb"
```

Rerun the **zipl** command so that the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLES11_SP1' (default)
Adding #2: IPL section 'FailsafeV1'
Adding #3: IPL section 'ipl'
Preparing boot device: dasda (0100).
Done.
```

These changes will be utilized the next time Linux is rebooted.

Note: Previous versions of this book recommended to list the minidisks to activate at boot time in the `zipl.conf` file. On SLES11 this approach does not work anymore. If you want to add DASD to the system, use the **dasd_configure** command instead of the two-step process using **chccwdev** and modifying `zipl.conf`.

If you want to check which hardware devices are activated at IPL, look in the `/etc/udev/rules.d/` directory at the `51-*` files.

7.5.10 Rebooting the system

You should now reboot the system to test the changes:

```
# reboot
```

After your system comes back, start a new SSH session.

7.5.11 Verifying the changes

You are now done customizing the cloner Linux image. **SSH** back into the cloner and check a few settings. Test the **vmcp** command with a CP command such as **QUERY NAMES**:

```
# vmcp q n
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC
VMSERVU  - DSC , VMSERVS  - DSC , TCPIP    - DSC , OPERSYMP - DSC
DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC , S11S1CLN - DSC
```

```
VSM     - TCPIP
```

Confirm that both of your swap spaces are operational:

```
# swapon -s
Filename                                Type            Size    Used
Priority
/dev/dasdf1                             partition       259956  0       -1
/dev/dasdg1                             partition       519924  0       -2
```

Verify that the NFS server is running:

```
# service nfsserver status
Checking for kernel based NFS server: idmapd                        running
 mountd                                                             running
 statd                                                              running
 nfsd                                                               running
```

Test mounting the SLES 11 SP1 install directory locally:

```
# ls /mnt
# mount localhost:/nfs/sles11sp1 /mnt
# ls /mnt
SLES-11-SP1-DVD-s390x-GM-DVD1.iso
```

This shows that you can mount the SLES 11 SP1 directory. Now unmount it:

```
# umount /mnt
```

Congratulations! You have installed and configured a SLES 11 SP1 Linux system onto the cloner. The next step is to install and configure the golden image.

8

Installing SLES 11 SP1 on the golden image

This chapter describes how to install SLES 11 SP1 onto the user ID `S11S1GLD`, which is referred to as the *golden image*. Chapters 4, 5, 6, and 7 must be completed before proceeding. The golden image is the copy of Linux that will be cloned. Normally the system is shut down and the user ID logged off because it is not recommended to clone a running Linux system.

In this example, the golden image is given two 3390-3s at minidisk addresses `100` and `101`. This allows for about 4.5 GB of disk space for each Linux system. If you want to increase that size, it is recommended that larger volumes such as 3390-9s be used, but that minidisks at addresses `100` and `101` still be used. This will allow the **`clone.sh`** script to continue to function.

8.1 Creating the S11S1GLD user ID

The golden image has a default memory size of 512 MB, and it is given class G privilege. It is given the following minidisks:

100 Half of the disk space for the golden image.

101 The other half of the disk space.

To accomplish this, perform the following steps:

1. Log on to MAINT
2. Edit the USER DIRECT file: add six new lines at the bottom of the file and create the following user directory entry. Set the 3390 disk labels to those appropriate for your system. In this example, the last third of ***UM63A2*** and the first third of a new 3390-9 volume, ***UM63A9***, are used:

```
==> x user direct c
====> bot
====> a 6
*
USER S11S1GLD LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 6677 3338 UM63A2 MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 0001 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
```

3. Add cylinder 0 of the new DASD volume to the dummy user ID $ALLOC$ so that the unused cylinders 0 do not show up as gaps.

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 610RES R
 MDISK A02 3390 000 001 UV6283 R
 MDISK A03 3390 000 001 UV6284 R
 MDISK A04 3390 000 001 UM6289 R
 MDISK A05 3390 000 001 UM6290 R
 MDISK A06 3390 000 001 UM6293 R
 MDISK A07 3390 000 001 UM6294 R
 MDISK A08 3390 000 001 UM63A2 R
 MDISK A09 3390 000 001 UM63A9 R
...
====> file
```

4. Run **DISKMAP** to check for overlaps and gaps. You should see a 501 and a 1-cylinder gap.

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
====> pre off
                                          0         500         501    GAP
--------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
--------------------  330 line(s) not displayed --------------------
====> quit
```

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

```
==> directxa user
```

```
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 44 disk pages
```

You have now defined the user ID that will contain the Linux golden image.

8.2 Creating the S11S1GLD parameter file

A SLES 11 SP1 parameter file is needed for this new user ID. You need to change the IP address (`HostIP` variable) and the host name (`Hostname` variable). In this example those are ***9.60.18.222*** and ***gpok222***.

Also, the cloner is used to provide the SLES 11 SP1 install directory using NFS by setting the `Install` variable.

Perform the following steps:

- Log on to `LNXMAINT`.
- Copy the `S11S1CLN` parameter file to one with a file name of `S11S1GLD` on the LNXMAINT 192 (D) disk:

  ```
  ==> copy s11s1cln parm-s11 d s11s1gld = =
  ```

- Edit the new file and set the networking values correctly. These changes set the golden image's IP address and host name, and also point to the new install server on the cloner (***9.60.18.223*** in this example).

  ```
  ==> x s11s1gld parm-s11
  ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
  HostIP=9.60.18.222 Hostname=gpok222.endicott.ibm.com
  Gateway=9.60.18.129 Netmask=255.255.255.128
  Broadcast=9.60.18.255 Layer2=0
  ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
  Nameserver=9.0.2.11 portname=whatever portno=0
  Install=nfs://9.60.18.223/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
  UseVNC=1  VNCPassword=12345678
  InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
  ```

When `S11S1GLD` is logged onto, the new parameter file will be accessible on the A (191) disk.

8.3 Updating 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.

Perform the following steps

- Log on to `AUTOLOG1`.
- Issue the **`ACCESS (NOPROF`** command so the PROFILE EXEC is not run:

  ```
  ==> acc (noprof
  ```

- Add a line to grant the `S11S1GLD` user ID access to the VSWITCH:

  ```
  ==> x profile exec
  ...
  ```

```
/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant s11s1cln'
'cp set vswitch vsw1 grant s11s1gld'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog s11s1cln'

'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 s11s1gld
Command complete
```

8.4 Installing the golden image

You should now be ready to begin the install the golden image. Linux will be installed onto the 100-101 minidisks. It will use 300-301 virtual disks for swapping. Most Linux user IDs described in this book will have two read/write minidisks and two VDISKS. Both swap disks are VDISKs, which means they are in z/VM virtual memory and thus provide fast access. Disk 300 is 256 MB and will act as a primary swap space. Only after it is full, disk 301, which is 512 MB, will be used. In a production environment there may also be another real DASD used as a swap device with the lowest priority—just in case.

To install the golden image, perform the following steps:

1. Log on to S11S1GLD. When you log on, you should see messages indicating that a virtual NIC has been created at address 0600 and that VDISKs 300 and 301 have been created

```
LOGON S11S1GLD
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:47:28 EDT WEDNESDAY 09/01/10
z/VM V6.1.0    2010-08-30 14:21

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
```

2. You are prompted to IPL Linux, but since you have not installed Linux yet, answer n:

```
Do you want to IPL Linux from minidisk 100? y/n
==> n
```

3. Before you install Linux, it is good to verify the resources. Verify that there are minidisks at virtual addresses 100 and 101 and virtual disks at addresses 300 and 301 using the **QUERY** command:

```
==> q 100-101
DASD 0100 3390 UM63A2 R/W       3338 CYL ON DASD  63A2 SUBCHANNEL = 0000
DASD 0101 3390 UM63A9 R/W       3338 CYL ON DASD  63A9 SUBCHANNEL = 0001
==> q 300-301
```

```
DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
```

4. Verify that you have a virtual OSA at addresses 600-602 with the **QUERY VIRTUAL OSA** command:

```
==> q osa
OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
     0600 DEVTYPE OSA         CHPID 00 OSD
     0600 MAC 02-00-01-00-00-03 CURRENT
     0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0003
     0601 DEVTYPE OSA         CHPID 00 OSD
     0601 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 0004
     0602 DEVTYPE OSA         CHPID 00 OSD
     0602 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
...
```

5. Use the **QUERY STORAGE** command to show that you have a 512 MB machine:

```
==> q stor
STORAGE = 512M
```

This shows that you have the resources necessary to install SLES 11 SP1.

8.4.1 Begin the SLES 11 SP1 installation

Follow these steps to begin the installation of S11S1GLD.

1. Run the **SLES11S1** exec. You should see many windows of questions and answers scrolling by. If you had used the default parameter file shipped with SLES 11 SP1, you would have had to answer all the networking questions manually. With the proper parameters set in the file S11S1GLD PARM-S11, the install process should proceed to where you access the install program using a VNC client:

```
==> sles11s1
NO FILES PURGED
RDR FILE 0001 SENT FROM S11S1GLD PUN WAS 0001 RECS 086K CPY  001 A NOHOLD
NOKEEP
RDR FILE 0002 SENT FROM S11S1GLD PUN WAS 0002 RECS 0009 CPY  001 A NOHOLD
NOKEEP
RDR FILE 0003 SENT FROM S11S1GLD PUN WAS 0003 RECS 167K CPY  001 A NOHOLD
NOKEEP
0000003 FILES CHANGED
0000003 FILES CHANGED
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32.12-0.7-default (geeko@buildhost) (gcc version 4.3.4
Ýgcc-4_
3-branch revision 152973¨ (SUSE Linux) ) #1 SMP 2010-05-20 11:14:20 +0200
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
  DMA      0x00000000 -> 0x00080000
  Normal   0x00080000 -> 0x00080000
Movable zone start PFN for each node
early_node_mapÝ1¨ active PFN ranges
    0: 0x00000000 -> 0x00020000
```

```
PERCPU: Embedded 11 pages/cpu @0000000001b43000 s12544 r8192 d24320 u65536
pcpu-alloc: s12544 r8192 d24320 u65536 alloc=16*4096
pcpu-alloc: Ý0¨ 00 Ý0¨ 01 Ý0¨ 02 Ý0¨ 03 Ý0¨ 04 Ý0¨ 05 Ý0¨ 06 Ý0¨ 07
pcpu-alloc: Ý0¨ 08 Ý0¨ 09 Ý0¨ 10 Ý0¨ 11 Ý0¨ 12 Ý0¨ 13 Ý0¨ 14 Ý0¨ 15
pcpu-alloc: Ý0¨ 16 Ý0¨ 17 Ý0¨ 18 Ý0¨ 19 Ý0¨ 20 Ý0¨ 21 Ý0¨ 22 Ý0¨ 23
pcpu-alloc: Ý0¨ 24 Ý0¨ 25 Ý0¨ 26 Ý0¨ 27 Ý0¨ 28 Ý0¨ 29 Ý0¨ 30 Ý0¨ 31
pcpu-alloc: Ý0¨ 32 Ý0¨ 33 Ý0¨ 34 Ý0¨ 35 Ý0¨ 36 Ý0¨ 37 Ý0¨ 38 Ý0¨ 39
pcpu-alloc: Ý0¨ 40 Ý0¨ 41 Ý0¨ 42 Ý0¨ 43 Ý0¨ 44 Ý0¨ 45 Ý0¨ 46 Ý0¨ 47
pcpu-alloc: Ý0¨ 48 Ý0¨ 49 Ý0¨ 50 Ý0¨ 51 Ý0¨ 52 Ý0¨ 53 Ý0¨ 54 Ý0¨ 55
pcpu-alloc: Ý0¨ 56 Ý0¨ 57 Ý0¨ 58 Ý0¨ 59 Ý0¨ 60 Ý0¨ 61 Ý0¨ 62 Ý0¨ 63
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 129280
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
TERM=dum
b                    HostIP=9.60.18.222 Hostname=gpok222.endicott.ibm.com
                     Gateway=9.60.18.129 Netmask=255.255.255.128
                     Broadcast=9.60.18.255 Layer2=0
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601
DataChannel=0.
0.0602               Nameserver=9.0.2.11 portname=whatever portno=0

Install=nfs://9.60.18.223/nfs/sles11sp1/SLES-11-SP1-DVD-s39
0x-GM-DVD1.iso       UseVNC=1  VNCPassword=12345678
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
PID hash table entries: 2048 (order: 2, 16384 bytes)
...
```

The install system or *starter system* should continue to boot. You should see the message:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.60.18.222, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.60.18.222:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***
```

2. You can use a VNC viewer through a Java-enabled browser, or a standalone VNC viewer. Connect to the VNC server (***9.60.18.222:1*** in this example). Enter the password specified in the parameter file (***12345678*** in this example).
3. You could disconnect from the 3270 session. However, messages to the console will be lost. It is recommended that you stay connected, but you may have to clear the window periodically (or the install process may be delayed waiting for the window to clear itself).

Now the graphical installation process should begin.

8.4.2 Beginning YaST installation

Perform the following steps to install SLES 11 SP1:

1. Choose your language and keyboard (**English US** in this example). Read the License Agreement, choose **I agree to the License Agreement** and click **Next**.
2. The Disk Activation window should appear. Choose **Configure DASD Disks**.

3. The DASD Disk Management window should appear: you will see all the DASDs available to `S11S1GLD`.
   a. Highlight each of the minidisks and VDISKs, 100, 101, 300, and 301 and click **Select or Deselect**.
   b. You should see a **Yes** appear next to them in the Sel. column on the left. Activate them by clicking **Perform Action** → **Activate**, as shown in Figure 8-1.

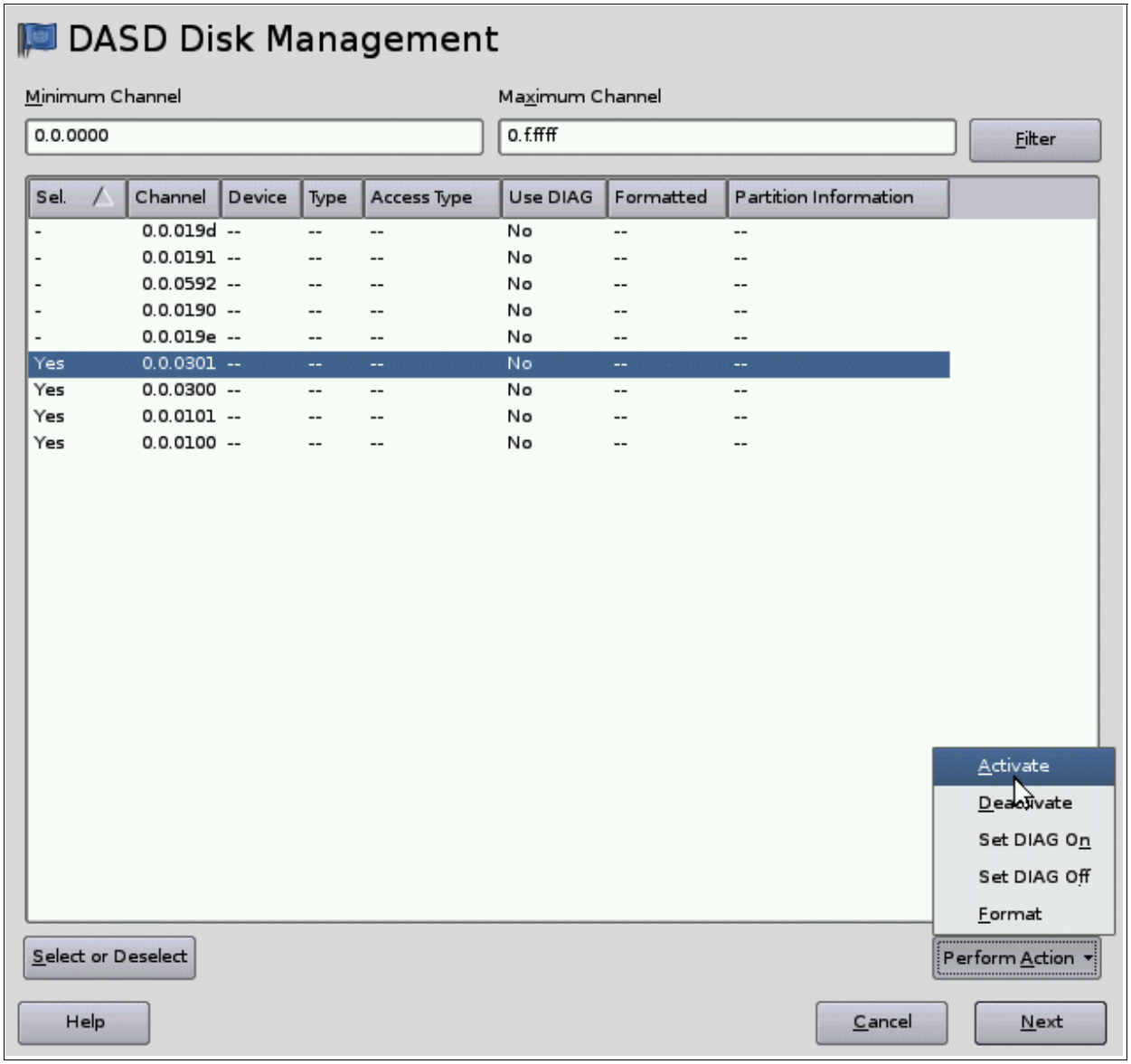


Figure 8-1 Activate the DASD available to S11S1GLD

   c. Disks 100 and 101 must be formatted so that Linux can use them. Deselect disks 300 and 301 using the **Select or Deselect** button, so that 100 and 101 remain selected. Now click **Perform Action** → **Format** as shown in Figure 8-2 on page 142.

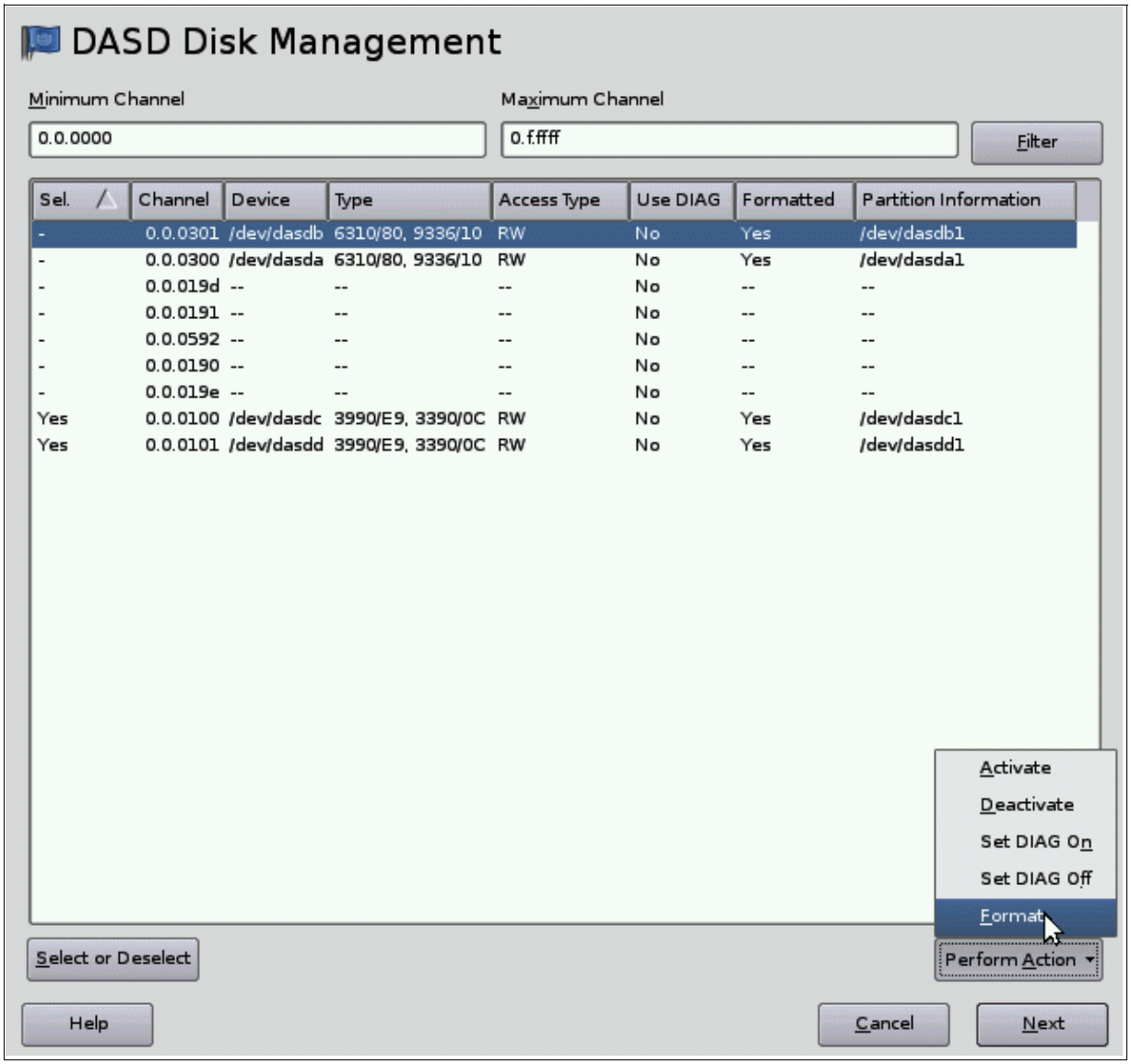


Figure 8-2 Format the DASD

4. You should see a window asking for 2 Parallel Formatted Disks. Click **OK**.
5. Click **Yes** to the question `Really format the following disks?`
6. A progress indicator window should appear displaying the progress of the parallel formats. This step can take 2-10 minutes depending on a number of factors.
7. When the formatting is complete, click **Next** in the DASD Disk Management window.
8. In the Disk Activation window, click **Next** again.
9. In the Installation Mode window, accept the default of **New installation** and click **Next**.
10. The Clock and Time Zone window will appear. Choose your region and time zone and click **Next**.
11. This will bring you to the Installation Settings window. Click **Partitioning** in the Overview tab. The Preparing Hard Disk window will appear.
12. Accept the default of **Custom Partitioning (for experts)** and click **Next**. The Expert Partitioner window will appear as shown in Figure 8-3 on page 143.

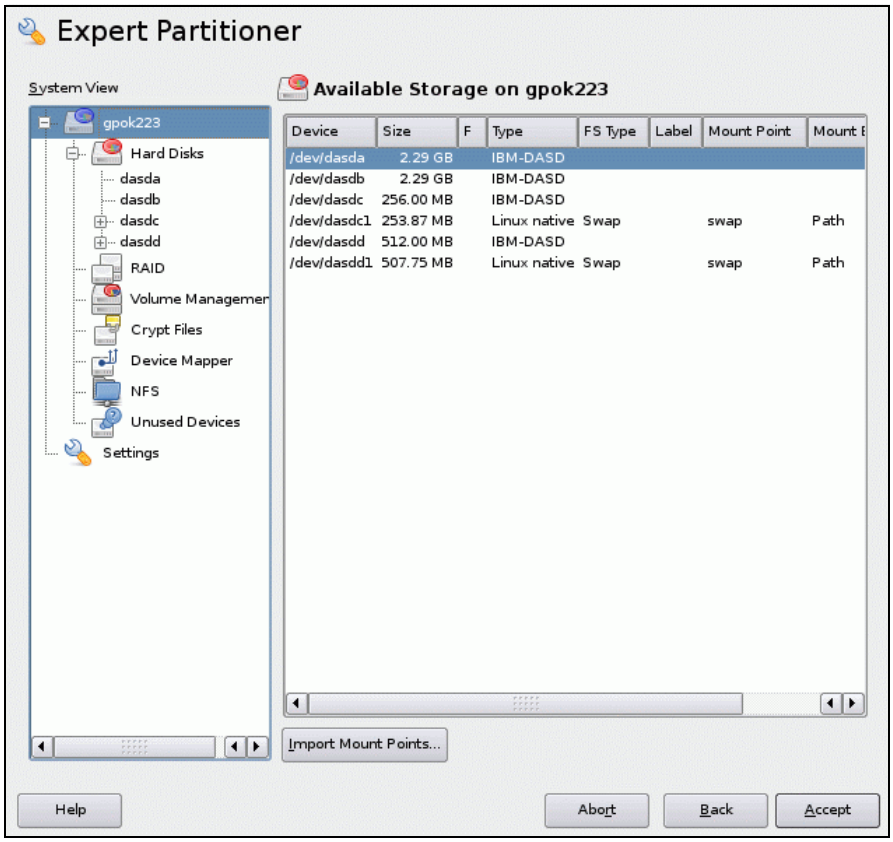


Figure 8-3 Disk partitioner - before customization

13. In this example, two partitions will be made on `/dev/dasda` and one partition will be made on `/dev/dasdb`.
    a. Click the plus sign (**+**) to the left of **Hard Disk**. It should show `dasda-dasdd` in the tree.
    b. Click **dasda** on the left and then **Add**. The window “Add Partition on /dev/dasda” will appear.
14. Accept the default **Custom size** and choose a size of 384 MB, as shown in Figure 8-4 on page 144. Click **Next**.

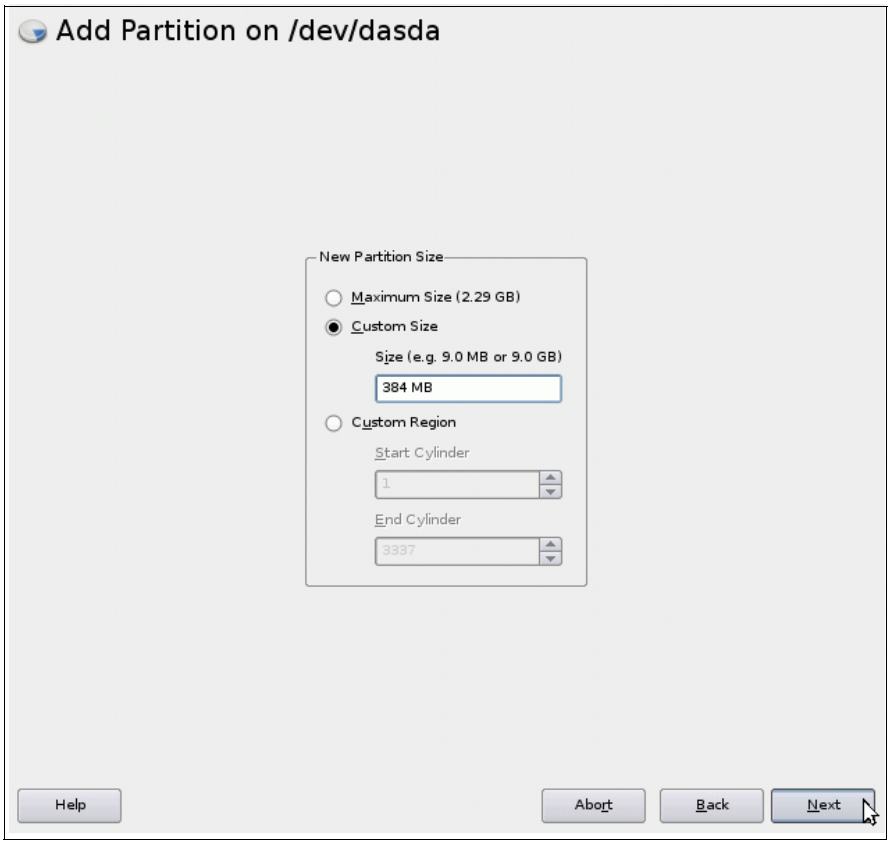


Figure 8-4 Add Partition on /dev/dasda

15. You should see a new window, “Add Partition on /dev/dasda.” If not set by default, perform the following steps:
    a. Accept the defaults of **Format Partition** and a File system of type **ext3** in the Formatting Options section.
    b. Accept the default **Mount Point** of **/** (root file system) in the Mounting Options section.
16. Click **Finish**. This creates the partition `/dev/dasda1`.
17. To create a second partition, click **Add** while `dasda` is selected in the left tree. The window “Add partition on /dev/dasda” will appear.
18. Click **Maximum Size**, then click **Next**.
19. In the new “Add partition on /dev/dasda” window, shown in Figure 8-5 on page 145, perform the following steps:
    a. Click **Do not format** in the Formatting Options section.
    b. Click **Do not mount partition** in the Mounting Options section.
    c. Click **Finish**. This creates an empty partition `/dev/dasda2`.

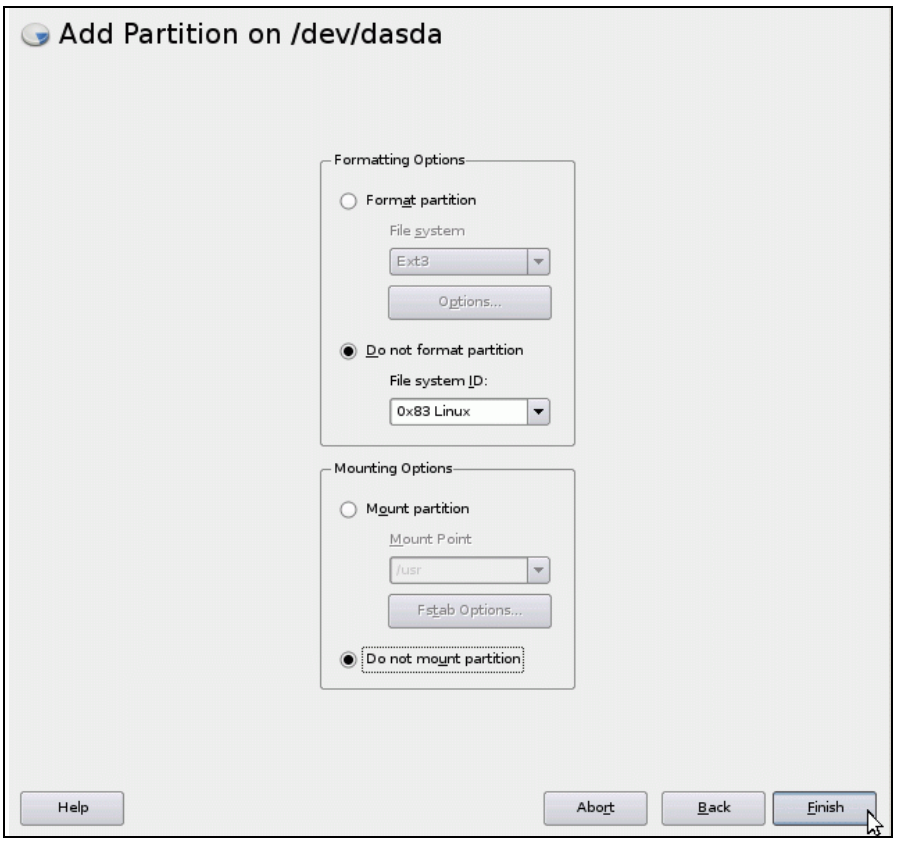


Figure 8-5 Add Partition on /dev/dasda

20. Create an empty partition, `/dev/dasdb1`, in the same fashion as you did to create `/dev/dasda2` (select **/dev/dasdb → Add → Maximum Size → Do not format/Do not mount partition**). Accept the default size that is to use all the space on the disk.
21. Now `/dev/dasda2` and `/dev/dasdb1` can be used to create a volume group. Click **Volume Management** as shown in Figure 8-6 on page 146.

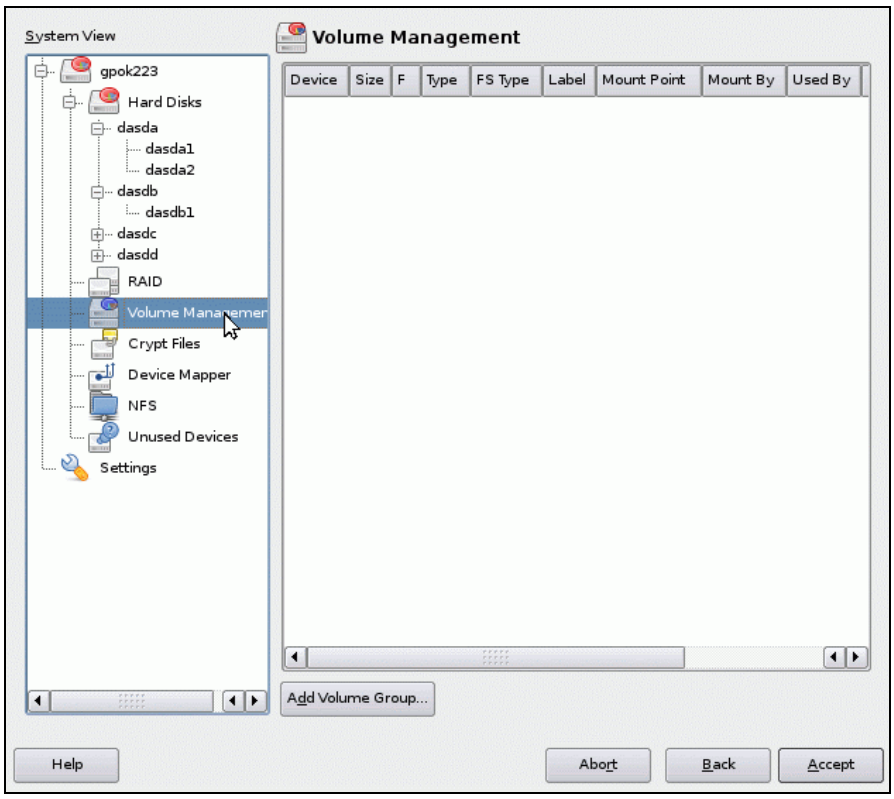


Figure 8-6 Creating logical volumes with Volume Management

22. Click **Add Volume Group**.
23. At the Add Volume Group window, set the volume group name to `system-vg`.
24. Click **Add All** to add both /dev/dasda2 and /dev/dasdb1 to the Selected Physical Volumes, as shown in Figure 8-7 on page 147. Click **Finish**.

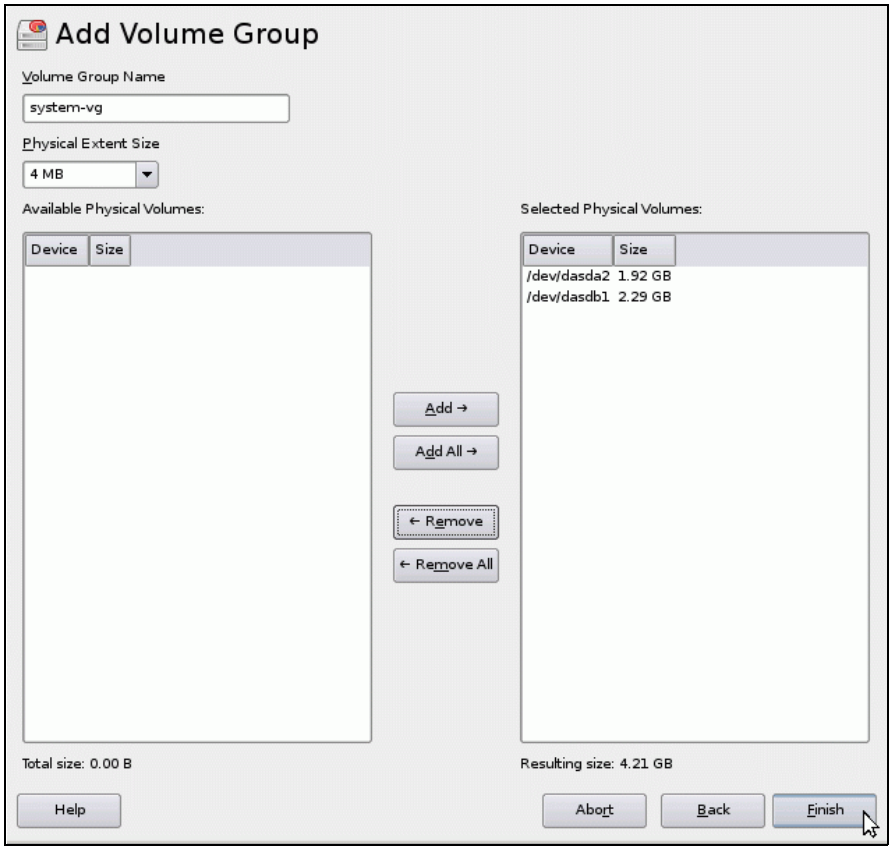


Figure 8-7 Add Volume Group

25. Back in Expert Partitioner, click the **plus sign** to the left of Volume Management. You should see the new system-vg volume group.
26. Click **system-vg** and the “Volume Group: /dev/system-vg” window will appear, as shown in Figure 8-8 on page 148. It shows that there are no logical volumes defined. Click **Add**.

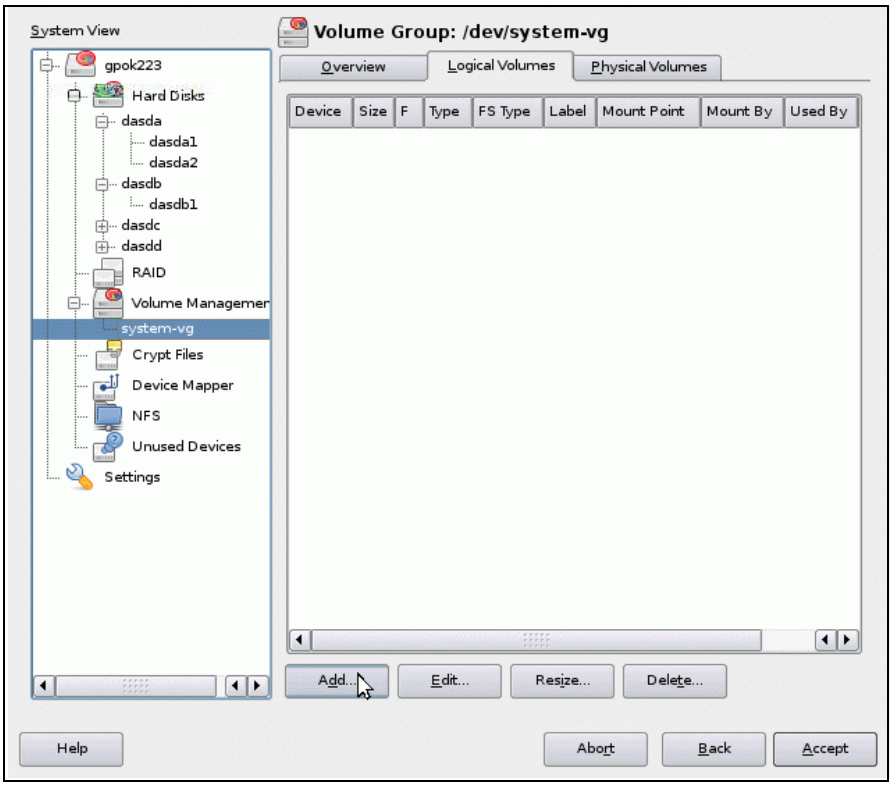


Figure 8-8 Volume Group view in Expert Partitioner

27. Enter `usr-lv` as the name for the new logical volume and click **Next**.
28. On the Add Logical Volume window, click **Manual Size** and enter a size of 2.5 GB. Accept the default of one stripe. Click **Next.**
29. Accept the defaults of **ext3** as the File System type and the Mount point of **/usr**, as shown in Figure 8-9 on page 149. Click **Finish**.

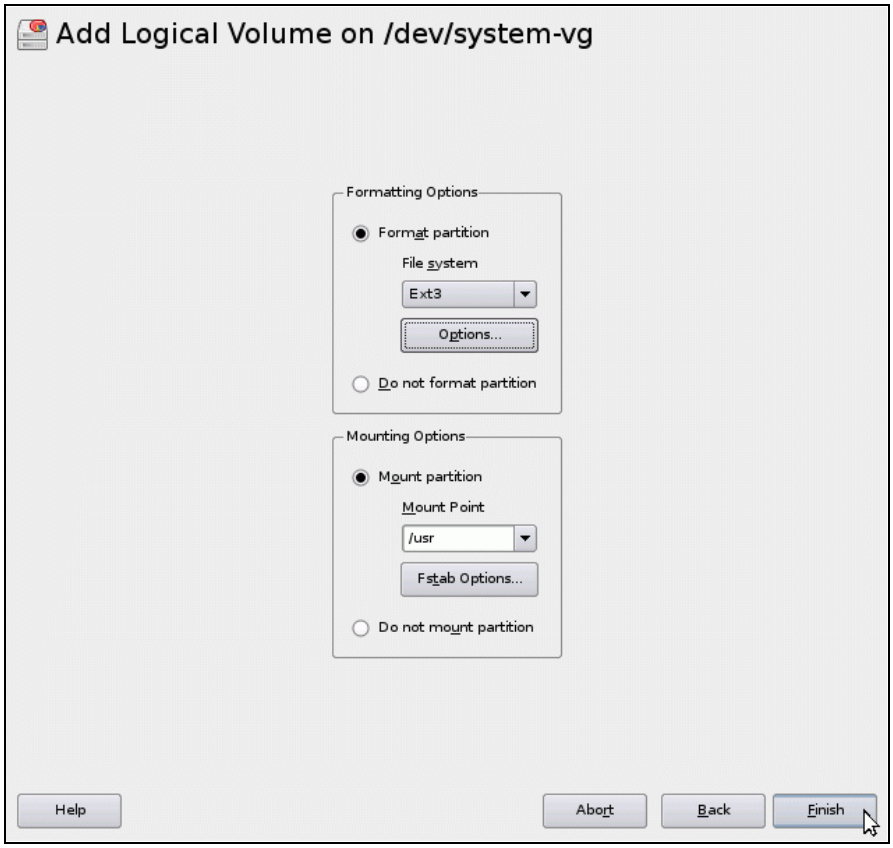


Figure 8-9 Adding a logical volume

30. Create three more logical volumes, similarly using Table 8-1.

Table 8-1 Logical volumes added to the system-vg volume group

Mount point	Logical volume name	Size
/usr/	usr-lv	2.5 GB
/var/	var-lv	512 MB
/opt/	opt-lv	384 MB
/tmp/	tmp-lv	384 MB

You may choose other file systems and sizes. See 2.3, “Disk planning” on page 10 for more discussion.

31. Back in the Expert Partitioner window, click the top object in the System View (***gpok222*** in this example). You should see output similar to Figure 8-10 on page 150. You do not have to format /dev/dasdc1 (vdev 300) and /dev/dasdd1 (vdev 301) because they were properly formatted as a Linux swap space by the SWAPGEN exec, and thus should be recognized as a swap space. Click **Accept**.

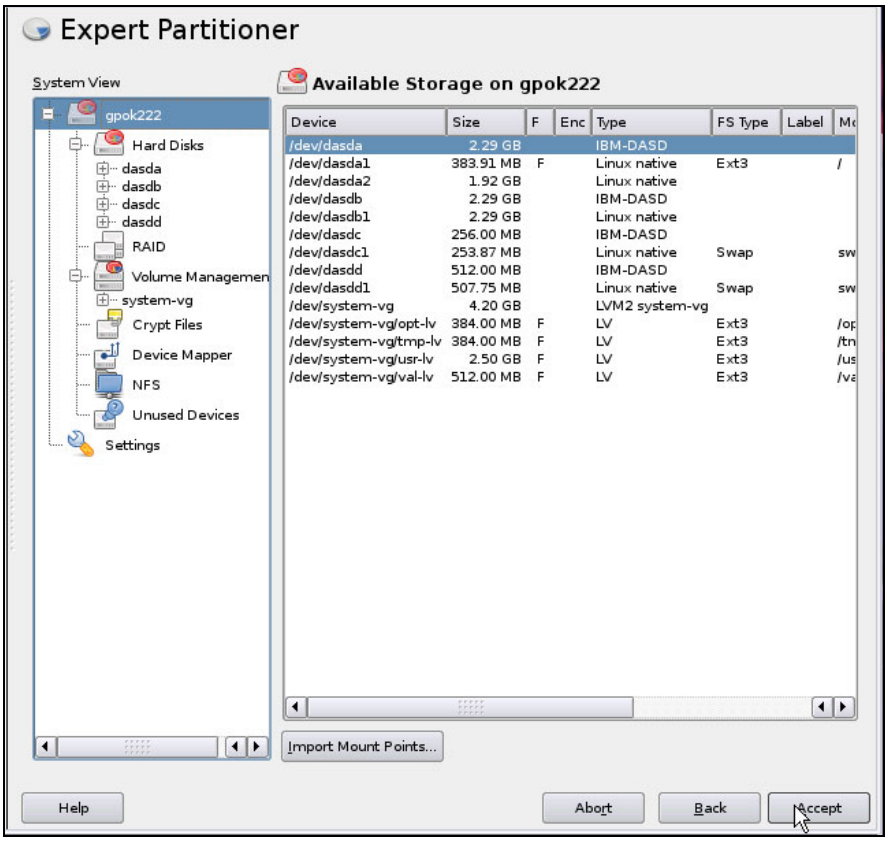


Figure 8-10 Disk partitioner - after customization

32. In the Installation Settings window, select **Software**. You may see a Disk Space Warning window. If so, click **OK**.
33. The Software Selection window opens. Leave **Base System, 32-Bit Runtime Environment**, **Help and Support Documentation** and **Minimal System** selected. Uncheck all other settings, as showed in Figure 8-11 on page 151. Click **OK**.

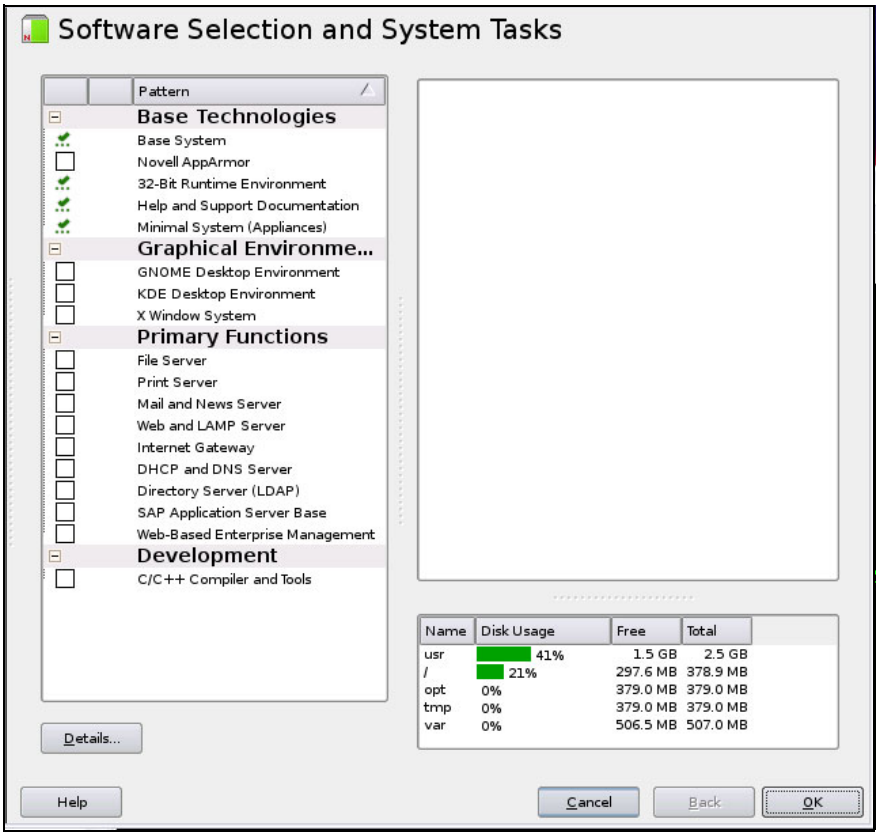


Figure 8-11 Software selection window

34. In the Expert tab move down to Default Runlevel and choose **3: Full multiuser with network**, as shown in Figure 8-12. Click **OK**. You will receive a VNC warning. Click **Yes**.

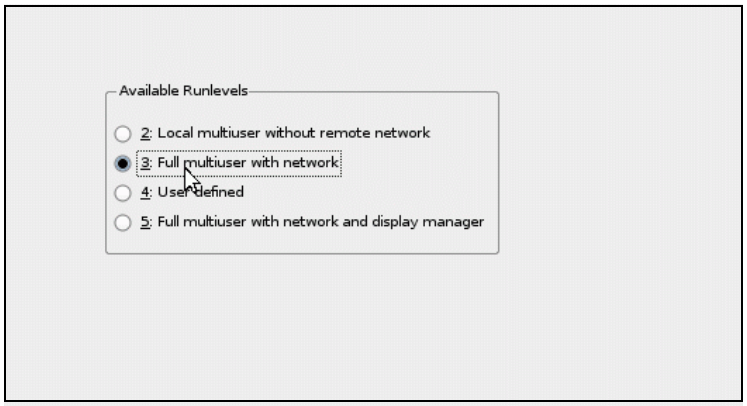


Figure 8-12 Default Runlevel

35. In the Languages window, select your primary and optional secondary languages and click **Accept**.
36. You are now ready to begin copying the RPMs onto your root file system. In the Installation Settings window, check the settings and click **Install**.
37. In the Confirm Installation window, click **Install**.

The SLES 11 SP1 system will be installed onto DASD. This should take about 5-20 minutes depending on a number of factors.

8.4.3 Rebooting the new Linux system from disk

After the first part of the installation completes, the Linux system reboots automatically. This time it boots from disk, not from the reader. You should see messages scrolling in the z/VM 3270 session. You will have to clear the window a number of times. The install program brings up a VNC server again to complete the installation.

```
...
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to 9.60.18.222, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.60.18.222:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)
...
```

8.4.4 Completing YaST2 installation

Go back to the same VNC client used for the first part of the installation. If it was a browser, either click **Log in Again** or click the browser's refresh button until another VNC login window appears. If you are using a VNC client then open that application again.

1. Log in using the same VNC password (`12345678` in this example).
2. In the "Password for root user" window, type the root password twice and click **Next**. *Do not forget* this password.
3. In the Hostname and Domain Name window, both Hostname and Domain Name are entered by the installer as they were specified in `S11S1GLD PARM-S11 file`. In this example ***gpok222*** is the host name and ***endicott.ibm.com*** is the domain name. Uncheck the **Change Hostname via DHCP** check-box. Click **Next**.
4. In the Network Configuration window you will see `Firewall is enabled`. Click the word **disable** to disable it, as shown in Figure 8-13 on page 153. All other values should be correct, so just click **Next**.

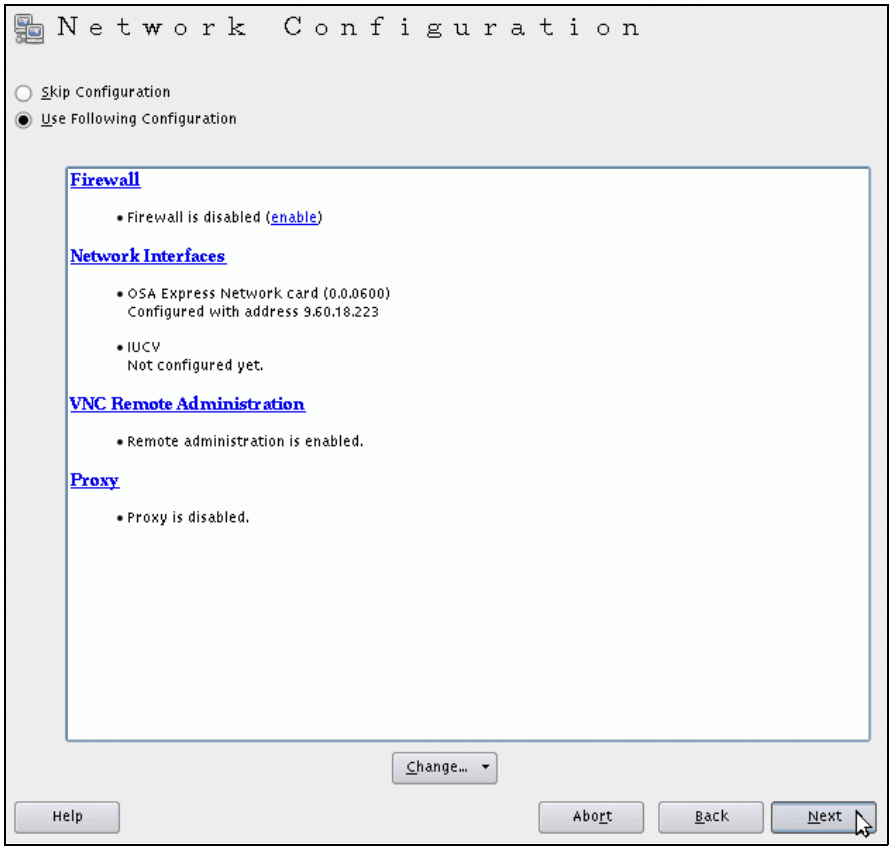


Figure 8-13 Disabling firewall in Network Configuration window

5. In the Test Internet Connection window, if you do not have Internet access, select **No, skip this test** and click **Next**.
6. In the Network Services Configuration window, accept the default and click **Next**. A certificate will be created.
7. In the User Authentication Method window, select **Local (/etc/passwd)** and click **Next**.
8. In the New Local User window, we recommend that you add at least one user so as to have a *non-root* ID on all cloned systems. In this example, the user `mikemac` was added. When you are done, click **Next**.
9. In the “Writing the system configuration” window, the SuSEconfig tool writes all your settings to disk.
10. The next window will be Release Notes. After reviewing the release notes, click **Next**.
11. In the Hardware Configuration window, choose **Skip Configuration** and click **Next**.
12. The last installation window is Installation Completed. Uncheck the Clone This System for Autoyast check box and click **Finish.**

The VNC session should end. Return to the 3270 session and you may have to clear the window a few times. Then you should see a login prompt. You are done installing Linux. You can disconnect from the 3270 session using the **`DISCONNECT`** command:

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

From this point forward, we recommend that you access your Linux systems using 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 19.

8.5 Configuring the golden image

Now you want to customize the golden image before cloning. The following high-level steps are recommended, though you may add or omit some steps. However, omitting steps can negatively affect the operation of the clone.sh script.

8.5.1 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 using the **vncserver** command.

SLES 11 SP1 configures a VNC connection, which starts by default. It is configured through the xinetd daemon. We recommend that you disable it. VNC will be configured to run only when needed.

Perform the following steps:

- Start an SSH session as root to the golden image.
- Edit the file `/etc/xinetd.d/vnc` and add a line with `disable=yes` in the sections service vnc1 and service vnchttpd1:

```
# cd /etc/xinetd.d
# vi vnc
service vnc1
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = nobody
        server          = /usr/bin/Xvnc
        server_args     = -noreset -inetd -once -from localhost -query
localhost -geometry 1024x768 -depth 16
        type            = UNLISTED
        port            = 5901
        disable         = yes
}

...
service vnchttpd1
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
```

```
        user            = nobody
        server          = /usr/bin/vnc_inetd_httpd
        server_args     = 1024 768 5901
        type            = UNLISTED
        port            = 5801
        disable         = yes
}
```

VNC is the only service that xinetd starts by default (this can be verified with the **chkconfig --list** command). You can turn off the xinetd service with the **service xinetd stop** command for this session and the **chkconfig command** at boot time:

```
# service xinetd stop
Shutting down xinetd:                                          done
# chkconfig xinetd off
```

Starting the VNC server manually

When you first start the VNC server, you are prompted to set a password. After it is set, this is the password that you will need to connect to it from a VNC client:

```
# vncserver
You will require a password to access your desktops.

Password: lnx4vm
Verify: lnx4vm
Would you like to enter a view-only password (y/n)? n
xauth:  creating new authority file /root/.Xauthority

New 'X' desktop is gpok222:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/gpok222:1.log
```

Stop the VNC server using the **-kill :1** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 22541
```

The icewm package allows for the IceWM, a window manager that is more usable than the Tiny Window Manager (twm) that VNC uses by default. We therefore recommend that you change to icewm. The package should already be installed on your system. Verify that the RPM was added:

```
# rpm -q icewm icewm-lite
icewm-1.2.36-1.35
icewm-lite-1.2.36-1.18
```

Change the window manager from twm to icewm in the file `/root/.vnc/xstartup`:

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

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

You may want to remove the `passwd` file so that the cloned system does not have the same password as the one you just entered. If so, use the command **`rm passwd`**. If you remove the password file, when a system is cloned, the password will be prompted for the first time that the VNC server is initialized.

Verifying VNC server configuration

Start the VNC server again:

```
# vncserver

New 'X' desktop is gpok222:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/gpok222:1.log
```

Start a VNC client, as shown on the left side of Figure 8-14. Enter the password. The resulting VNC session is shown on the right side of Figure 8-14.

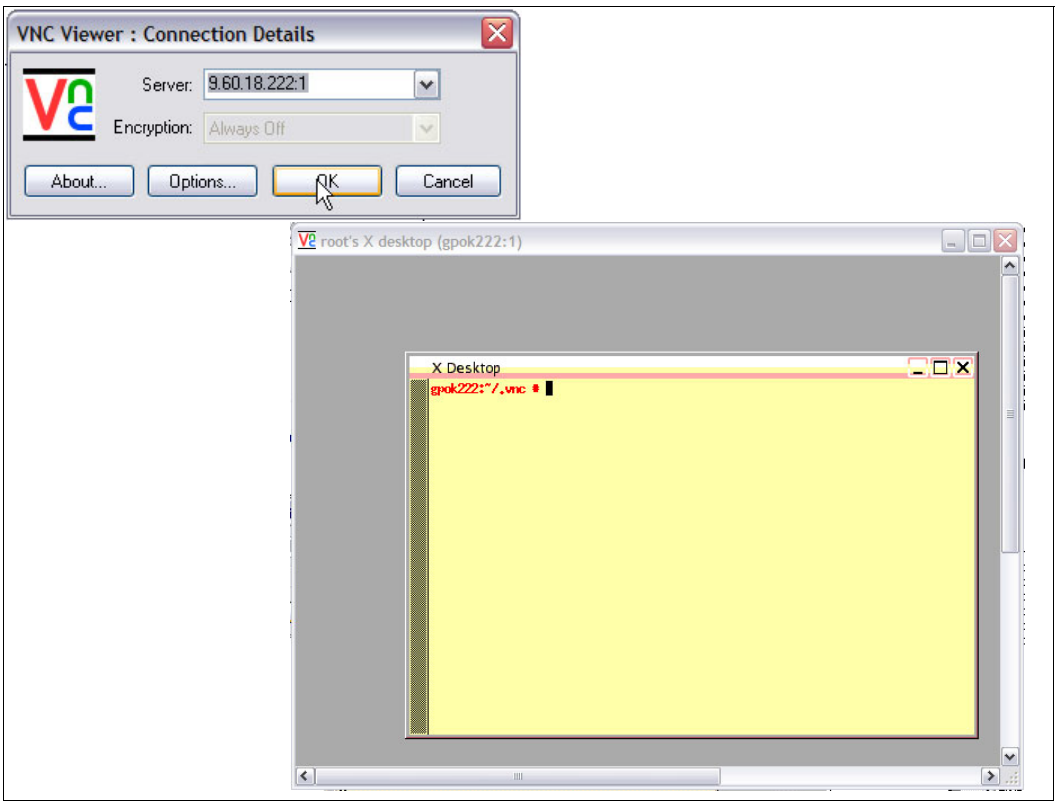


Figure 8-14 VNC session

Note that the VNC server will not be started across reboots. When you need a graphical environment, you either have to start **`vncserver`** manually (recommended), or you can revert to **`xinetd`** starting it.

Unless you need a graphical environment, you can again stop the VNC server using the **`-kill :1`** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 22601
```

Do not forget to remove the password file, so that your password from the golden image is not distributed to every cloned image (unless it is your intent).

8.5.2 Preparing for Online Update

This step is recommended if you have Internet access.

Note: The system used for the writing of this book did not have Internet access, so the following steps have not been tested for SLES 11 SP1.

Before you can use Online Update (recommended), online sources have to be configured. This is done through the Novell Customer Center Configuration. To configure the Customer Center, a web browser is needed. For this reason, a VNC server session must be started.

A graphical environment is recommended for this step. Start a VNC viewer. In a terminal session, start YaST with the **yast2** command and choose **Software** → **Online Update Configuration**.

8.5.3 Turning off unneeded services

There are a number of services that are started in a SLES 11 SP1 minimal system. They can be viewed using the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog             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
haldaemon               0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer            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
network-remotefs        0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
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
rpcbind                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
smartd                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
splash                  0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early            0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

In order to keep the golden image as lean as possible in terms of processor usage, some of these services can be turned off with the **chkconfig** command:

```
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
```

You may choose to leave these services on, or turn others off. You can review which services are now configured to start in runlevel 3 with the following command:

```
# chkconfig -l | grep 3:on
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog             0:off  1:off  2:on   3:on   4:off  5:on   6:off
haldaemon               0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer            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
nfs                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

This shows the services that will run in the default runlevel 3.

8.5.4 Enabling the cmm module

The cmm module allows Linux to do cooperative memory management (aka CMM1) with z/VM. This feature must be enabled on both the Linux and z/VM sides.

To enable it, edit the file `/etc/sysconfig/kernel` and add `cmm` to the variable `MODULES_LOADED_ON_BOOT` (around line 30):

```
# cd /etc/sysconfig
# vi kernel
...
## Type:               string
## ServiceRestart:     boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
#
MODULES_LOADED_ON_BOOT="cmm"
...
```

Save the file. The cmm module should be loaded after your system is rebooted.

8.5.5 Applying service - online update

If you have a SuSE Maintenance web account, you can use it to retrieve the latest patches for SLES 11 SP1. Because many of these patches contain security and bug fixes, it is recommended that you apply the patches for the golden image so that it is up to date. Subsequently all the servers you clone after the golden image will also be up to date.

Section 8.5.2, “Preparing for Online Update” on page 157 must be completed before proceeding.

The system used for the writing of this book did not have Internet access, so this part was not tested at all. You may choose which packages to install and which not. We highly recommend to download security patches.

When you finish here, do not reboot yet. You will reboot shortly after some boot configuration changes are made.

8.5.6 Configuring /etc/inittab

If you did not change the default runlevel from 5 to 3 during the installation process, now would be a good time to do so. By default, SLES 11 boots into runlevel 5, which is designed for a graphical environment. To keep the golden image as lean as possible, we recommend that this be reset to runlevel 3. This is set in the `/etc/inittab` file with the variable `initdefault`.

Linux reboots when a Ctrl-Alt-Del key sequence is trapped. This key sequence is simulated by z/VM when it issues a **`SIGNAL SHUTDOWN`** command. Rather than rebooting, it might be better that your system halts (shuts down).

Change **`shutdown -r`** to **`shutdown -h`**:

```
# cd /etc
# vi inittab
...
# The default runlevel is defined here
id:3:initdefault:
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

This change will be picked up when the system is rebooted.

8.5.7 On Demand Timer patch

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 many processor cycles as the number of virtual servers goes up. A rule of thumb on System z is to turn off this timer unless the server has a heavy, constant workload. Otherwise, even an idle Linux image never goes idle from a z/VM point of view.

In the past, there was the On Demand Timer patch needed on System z which solved this problem (disabled the timer pop-up). It was System z-specific code. Later on, people on x86 also discovered that the 10 ms timer can also have drawbacks, mainly because it reduced battery lifetime. So generic timer code was developed. This code is enabled by default and makes sure that an idle Linux guest stays idle also from the z/VM point of view.

With the new code, system control was also dropped. With system control it was possible to enable or disable the timer at runtime. The current code just works and it is enabled by default. The only interface to it is the kernel parameter `nohz=` that makes it possible to switch off the NOHZ code at boot time.

8.5.8 Configuring SSH keys

SSH sessions are typically authenticated using passwords typed in on the keyboard. With SSH key-based authentication sessions can be authenticated using public and private keys so that no password is needed. To accomplish this, the following must be true:

- The SSH server system must have the client's public key.

- The SSH client must send its private key.
- The keys must match cryptographically.

SSH key-based authentication can be set up from the cloner (client) to the virtual servers. If the preceding requirements are met, then key-based authentication will work to the cloned virtual servers.

To set up key-based authentication from the cloner to the golden image (and consequently from cloner to all cloned images), perform the steps that follow. **(*gpok222*)** and the cloner **(*gpok223*),** so prefixes are used in the following examples.

- On the golden image, create the directory `/root/.ssh` and set the permission bits to octal 700 with the **chmod** command:

```
gpok222: # cd /root
gpok222: # mkdir .ssh
gpok222: # chmod 700 .ssh
```

- Start an SSH session as root to the cloner. The IP address is ***9.60.18.223*** in this example.
- Create a new key of type DSA in root's `.ssh/` directory:

```
gpok223:~ # cd .ssh
gpok223:~/.ssh # ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
30:5e:02:e4:ee:86:55:e9:8f:05:97:4e:3e:7f:25:c3 root@gpok223
The key's randomart image is:
+--[ DSA 1024]----+
|   .o            |
|   . . . .       |
|    . B =        |
|   . + @   .     |
|    o o S   E .  |
|   +   + o   +   |
|  . o . . . .    |
|   .       .     |
|                 |
+-----------------+
```

- Copy the new public key to the `/root/.ssh/` directory that you just created on the golden image (***9.60.18.222*** in this example) with the special file name of `authorized_keys`:

```
gpok223:~/.ssh # scp id_dsa.pub 9.60.18.222:/root/.ssh/authorized_keys
The authenticity of host '9.60.18.222 (9.60.18.222)' can't be established.
RSA key fingerprint is f1:6e:3e:70:f4:e3:c4:58:a5:e8:4a:0d:f6:b4:44:c8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.60.18.222' (RSA) to the list of known hosts.
Password: lnx4vm
id_dsa.pub                                    100%  602     0.6KB/s   00:00
```

You should now be able to get an SSH session to the golden image without needing a password, because the cloner is recognized as an authorized host:

```
gpok223: # ssh 9.60.18.222
```

```
Last login: Mon Nov 30 07:57:27 2009 from sig-9-145-170-94.de.ibm.com
gpok222:~ # exit
```

If you get an SSH session without having to supply a password, as in the preceding example, it shows that key-based authentication is working.

8.5.9 Modifying zipl.conf

In previous versions of this book, this section described how to add disks that should be brought online during IPL. The `dasd=` parameter in the `/etc/zipl.conf` file was used for that. In SLES 11 this parameter is no longer supported. Instead, the **`dasd_configure`** script is used for adding disks to the system.

Two changes are recommended to the default `zipl.conf` file:

- The parameters `vmpoff=LOGOFF` and `vmhalt=LOGOFF` have the effect of logging the z/VM user ID off when Linux is shut down. This can be convenient for shutting the z/VM system down more efficiently and also for getting a refreshed 3270 emulator session.
- The `timeout` value is modified to 3 seconds. When SLES 11 SP1 boots, the default time that is allowed to enter a menu command is 10 seconds. Because the default value is most commonly used, these 10 seconds delay the starting of Linux. It is backed up to three seconds.

Make a backup copy of the file and add the string `vmpoff=LOGOFF` to the parameter line in the `[Linux]` section and set `timeout = 3` in the menu section:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf   // add the dasd= string
# Modified by YaST2. Last modification on Wed Sep  1 12:37:42 EDT 2010
[defaultboot]
defaultmenu = menu

###Don't change this comment - YaST2 identifier: Original name: linux###
[SLES11_SP1]
    image = /boot/image-2.6.32.12-0.7-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.32.12-0.7-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 vmpoff=LOGOFF
vmhalt=LOGOFF TERM=dumb"

###Don't change this comment - YaST2 identifier: Original name: failsafe###
[FailsafeV1]
    image = /boot/image-2.6.32.12-0.7-default
    target = /boot/zipl
    ramdisk = /boot/initrd-2.6.32.12-0.7-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb
x11failsafe"


:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 3
    1 = SLES11_SP1
```

```
    2 = FailsafeV1
    3 = ipl

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
    image = /boot/image
    target = /boot/zipl
    ramdisk = /boot/initrd,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb"
```

Now, run the **zipl** command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLES11_SP1' (default)
Adding #2: IPL section 'FailsafeV1'
Adding #3: IPL section 'ipl'
Preparing boot device: dasda (0100).
Done.
```

These changes are utilized the next time Linux is rebooted.

8.5.10 Cleaning temporary files

Now is the best time to clean temporary files from the golden image. If they stay there they will be copied to each clone and occupy space there. There are a number of directories that can be cleaned up and the history of the commands executed can be cleared.

- Create a script, `cloneprep.sh`, that can be run to prepare for cloning:

  ```
  # cd /usr/local/sbin
  # vi cloneprep.sh
  #!/bin/bash
  rm -fr /tmp/.ICE-unix
  rm -fr /tmp/.X11-unix
  rm -rf /tmp/*
  rm -f /var/log/YaST2/*
  history -c
  ```

  You may choose different steps to prepare for cloning.

- Make the script executable with the **chmod +x** command:

  ```
  # chmod +x cloneprep.sh
  ```

- Call the script interactively:

  ```
  # cloneprep.sh
  ```

The system should now be cleaned for cloning.

8.5.11 Other configuration changes

You may consider other configuration changes. Of course you can take an interative approach: start with this set of changes, clone some Linux images and test, then bring the golden image back up, make more changes, and reclone.

Whether you are on the first pass of configuration or not, refer to the following sections to consider other changes for performance and availability related issues:

- 11.5, “Setting up Memory Hotplug” on page 215
- 11.6, “Utilizing the cpuplugd service” on page 218
- 11.7, “Hardware cryptographic support for OpenSSH” on page 223

8.5.12 Rebooting the system and verifying changes

You are now done customizing the golden Linux image. Now **reboot** to test your changes:

```
# reboot
Broadcast message from root (pts/0) (Mon Nov 30 08:51:49 2009):

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

When the system comes back up, you should verify the changes that you made.

1. SSH back into the golden image and check a few settings.
2. Use the **df -h** command to display your file systems:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           372M  104M  249M  30% /
devtmpfs              248M  120K  247M   1% /dev
tmpfs                 248M     0  248M   0% /dev/shm
/dev/mapper/system--vg-opt--lv
                      372M   17M  337M   5% /opt
/dev/mapper/system--vg-tmp--lv
                      372M   17M  337M   5% /tmp
/dev/mapper/system--vg-usr--lv
                      2.5G  1.2G  1.2G  50% /usr
/dev/mapper/system--vg-val--lv
                      504M   55M  424M  12% /var
```

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

```
# swapon -s
Filename                                Type            Size    Used
Priority
/dev/dasdc1                             partition       259956  0       -1
/dev/dasdd1                             partition       519924  0       -2
```

   Device with higher priority will be used first.

4. Shut down your golden image from the SSH session (to clone Linux it should be shut down):

```
# shutdown -h now
```

Congratulations! You have now successfully installed the golden image. This image will normally be shut down.

9

Cloning SLES 11 SP1

This chapter focuses on cloning the golden image. Examples are given for cloning manually and through the use of a script. The following steps are described:

9.1 Formatting DASD for minidisks

In this section, DASD is formatted to become minidisks for the cloned images. The **CPFMTXA** command can be used to format one DASD at a time, but the CPFORMAT exec is a wrapper around **CPFMTXA** that allows the formatting of multiple DASD.

To have access to enough DASD to define four more user IDs with two 3390-3 volumes each, three more 3390-9 volumes must be formatted. In the examples used in this book, the two volumes that will be used are at addresses 63AA and 63AB. Consult your worksheets on 2.7.2, “z/VM DASD worksheet” on page 17.

1. Log on to a 3270 session as MAINT.
2. Query the devices that will be used for LINUX01-LINUX04. In this example they are 63AA-63AB.

   ```
   ==> q 63aa-63ab
   DASD 63AA DM63AA , DASD 63AB DM63AB
   ```

3. Attach the three volumes that will be used for LINUX01-LINUX04. In this example, it is the DASD at addresses 63A9 through 63AB:

   ```
   ==> att 63aa-63ab *
   63AA-63AB ATTACHED TO MAINT
   ```

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

   ```
   ==> cpformat 63aa-63ab as perm

   Format the following DASD:
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    63AA MAINT    63AA 3390  DM63AA 63AA          0      10017
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    63AB MAINT    63AB 3390  DM63AB 63AB          0      10017

   WARNING - this will destroy data!
   ARE YOU SURE you want to format the DASD as PERM space (y/n)?
   y
   ...
   DASD successfully formatted: UM63AA UM63AB
   63AA-63AB DETACHED
   63AA-63AB ATTACHED TO MAINT

   DASD status after:
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    63AA MAINT    63AA 3390  UM63AA 63AA          0      10017
   MAINT    63AB MAINT    63AB 3390  UM63AB 63AB          0      10017
   ```

Detach the three volumes from MAINT using the **DETACH** command:

```
==> det 63aa-63ab
63AA-63AB DETACHED
```

Attach the newly formatted DASD to SYSTEM so they can be used for minidisks:

```
==> att 63aa-63ab system
DASD 63AA ATTACHED TO SYSTEM UM63AA
DASD 63AB ATTACHED TO SYSTEM UM63AB
```

The two DASD volumes are now available to be used for minidisks in the USER DIRECT file. They will also be available after the next IPL because their new labels match the pattern specified by the **User_Volume_Include UM6*** statement in the SYSTEM CONFIG file.

9.2 Defining a new user ID for a virtual server

In this section you will define a new user ID, LINUX01.

1. Log on to MAINT and edit the USER DIRECT file to add a new user ID, LINUX01. Define it with the same sized 100 and 101 minidisks so that the golden image can be cloned to it:

   ```
   ==> x user direct c
   ```

2. Go to the bottom of the file (**bot**) and add six lines (**a6**). In this example the user ID will be LINUX01 with a password of LNX4VM. The two minidisks with virtual addresses 100 and 101 are taken from the last two thirds of the 3390-9 with a label of UM63A9:

   ```
   *
   USER LINUX01 LNX4VM 256M 1G G
    INCLUDE LNXDFLT
    OPTION APPLMON
    MDISK 100 3390 3339 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
    MDISK 101 3390 6677 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
   ```

3. Again check for gaps and overlaps with the **DISKMAP** command. You can use the **ALL** subcommand with the logical OR operator "|" to check for both strings. You should see two gaps of 501 cylinders and 1 cylinder.

   ```
   ==> diskmap user
   ==> x user diskmap
   ====> pre off
   ====> all /gap/|/overlap/
   --------------------  4  line(s) not displayed --------------------
                                              0         500         501    GAP
   --------------------  6  line(s) not displayed --------------------
                                              0           0           1    GAP
   --------------------  338  line(s) not displayed --------------------
   ====> quit
   ```

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

   ```
   ==> directxa user
   z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
   EOJ DIRECTORY UPDATED AND ON LINE
   HCPDIR494I User directory occupies 44 disk pages
   ```

The new Linux user ID has now been defined.

9.2.1 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 automatically start the Linux user IDs:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
==> x profile exec f   // add two lines
/*************************/
/*  Autolog1 Profile Exec  */
/*************************/
'cp xautolog tcpip'                 /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 300'        /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant s11s1cln'
'cp set vswitch vsw1 grant s11s1gld'
'cp set vswitch vsw1 grant linux01'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog s11s1cln'
'cp xautolog linux01'
'cp logoff'                         /* logoff when done */
====> file
```

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

```
==> set vswitch vsw1 grant linux01
Command complete
```

The user ID LINUX01 should now have permission to attach its NIC to the VSWITCH.

9.2.2 Creating a parameter file for the new LINUX ID

For each Linux guest you want to clone, you need to create a parameter file. This file specifies many of the installation parameters. It will be used both when cloning to this user ID and when installing SLES 11 SP1 manually.

1. Log on to LNXMAINT.
2. Copy an existing parameter file to a new one with the file name equal to the new Linux user ID:

   ```
   ==> copy s11s1gld parm-s11 d linux01 = =
   ==> x linux01 parm-s11 d
   ```

3. Edit the new parameter file as you did for S11S1GLD (see “Creating the S11S1GLD parameter file” on page 137). If the new Linux is going to be on the same network as the cloner, you will likely only have to change two variables: the IP address and the DNS name. In this example, the IP address is set to 9.60.18.224 and the DNS name to gpok224.endicott.ibm.com:

   ```
   ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
   ```

```
HostIP=9.60.18.224 Hostname=gpok224.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.223/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

4. Log off of LNXMAINT and log on to LINUX01.

5. Watch for error messages as LINUX01 is logged on. Answer N to the question "Do you want to IPL Linux from minidisk 100".:

```
LOGON LINUX01
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 08:39:10 EST TUESDAY 12/01/09
z/VM V6.1.0    2009-11-19 13:47

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
...
```

6. Verify that the new Linux user ID has an NIC at addresses 600-602:

```
==> q osa
OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
     0600 DEVTYPE OSA         CHPID 00 OSD
     0600 MAC 02-00-01-00-00-03 CURRENT
     0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0003
     0601 DEVTYPE OSA         CHPID 00 OSD
     0601 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 0004
     0602 DEVTYPE OSA         CHPID 00 OSD
     0602 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
```

7. Use the **QUERY DASD** command to verify that the minidisks at 100 and 101 and the VDISKs at 300 and 301 are read-write:

```
==> q da
DASD 0100 3390 UM63A9 R/W       3338 CYL ON DASD  63A9 SUBCHANNEL = 0000
DASD 0101 3390 UM63A9 R/W       3338 CYL ON DASD  63A9 SUBCHANNEL = 0001
DASD 0190 3390 610RES R/O        107 CYL ON DASD  6280 SUBCHANNEL = 0009
DASD 0191 3390 UM6289 R/O        300 CYL ON DASD  6289 SUBCHANNEL = 000C
DASD 019D 3390 UV6283 R/O        146 CYL ON DASD  6283 SUBCHANNEL = 000A
DASD 019E 3390 UV6283 R/O        250 CYL ON DASD  6283 SUBCHANNEL = 000B
DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
DASD 0592 3390 UV6284 R/O         70 CYL ON DASD  6284 SUBCHANNEL = 000D
```

8. Log off of LINUX01. To clone to a user ID, it should be logged off so the target disks can be linked read-write.

You should now be ready to clone to this new user ID.

9.3 Cloning a virtual server manually

Before using the shell script clone.sh to clone a server, you may want to clone a server manually to better understand the process. There are many ways to clone Linux under z/VM. The steps in this section are just one way to do it. The following assumptions are made based on what you have done so far:

- The source (golden) Linux image is on the user ID S11S1GLD on minidisks 100 and 101.
- The target user ID, LINUX01 in this example, has identically sized minidisks.
- The **vmcp** command is available to the cloner, S11S1CLN, to issue z/VM CP commands.
- The z/VM **FLASHCOPY** command can be used but if you do not have that support, the Linux dasdfmt and **dd** commands can also be used.

Given these assumptions, one set of steps that can be used to clone a system is as follows:

1. Link the source disks read-only.
2. Link the target disks read-write.
3. Copy the source to the target disk using **FLASHCOPY** or the Linux **dasdfmt** and **dd** commands.
4. Mount the newly copied root file system.
5. Modify the networking information on the target system.
6. Detach the target disks.
7. IPL the target system.
8. Modify the SSH keys on the target system.

The following sections describe these steps in detail.

Linking the source disks read-only

Start an SSH session as root to the cloner, S11S1CLN.

The source minidisks at virtual addresses 100 and 101 are on the golden image, S11S1GLD. They are linked read-only as virtual addresses 1100 and 1101 using the **RR** parameter to the CP **LINK** command:

```
# vmcp link s11s1gld 100 1100 rr
# vmcp link s11s1gld 101 1101 rr
```

Linking the target disk read-write

The target minidisks, also at addresses 100 and 101, are on the cloner, LINUX01. They are linked multi-read (read-write if no other user ID has write access) using the **MR** parameter as virtual device 2100 and 2101 using the CP **LINK** command:

```
# vmcp link linux01 100 2100 mr
# vmcp link linux01 101 2101 mr
```

Copying the source to the target disk using FLASHCOPY

The two disks are copied using the CP **FLASHCOPY** command:

```
# vmcp flashcopy 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 3337 TO 2100 0 3337
# vmcp flashcopy 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 3337 TO 2101 0 3337
```

Note: If you do not have **FLASHCOPY** support, you can use the Linux **dasdfmt** and **dd** commands. You must first enable the source and target disks using the **chccwdev -e** command, then determine the device name using the **lsdasd command**:

```
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
# chccwdev -e 1101
Setting device 0.0.1101 online
Done
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
# chccwdev -e 2101
Setting device 0.0.2101 online
Done
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2122MB    543240
0.0.0102   active      dasdc     94:8    ECKD  4096   2347MB    600840
0.0.0103   active      dasdd     94:12   ECKD  4096   2347MB    600840
0.0.0104   active      dasde     94:16   ECKD  4096   2347MB    600840
0.0.0105   active      dasdf     94:20   ECKD  4096   2347MB    600840
0.0.0300   active      dasdg     94:24   FBA   512    256MB     524288
0.0.0301   active      dasdh     94:28   FBA   512    512MB     1048576
0.0.1100   active      dasdi     94:32   ECKD  4096   2347MB    600840
0.0.1101   active      dasdj     94:36   ECKD  4096   2347MB    600840
0.0.2100   active      dasdk     94:40   ECKD  4096   2347MB    600840
0.0.2101   active      dasdl     94:44   ECKD  4096   2347MB    600840
```

In this example, the source devices are /dev/dasdi and /dev/dasdj, and the target devices are /dev/dasdk and /dev/dasdl. Format the target disks using the **dasdfmt** command, then copy them using the **dd** command using a block size of 4 K (4096) bytes:

```
# dasdfmt -b 4096 -y -f /dev/dasdk
...
# dasdfmt -b 4096 -y -f /dev/dasdl
...
# dd if=/dev/dasdi of=/dev/dasdk bs=4096
600840+0 records in
600840+0 records out
# dd if=/dev/dasdj of=/dev/dasdl bs=4096
600840+0 records in
600840+0 records out
```

The golden image should now be copied to the target disks. Disable the disks:

```
# chccwdev -d 1100
```

```
# chccwdev -d 1101
# chccwdev -d 2100
# chccwdev -d 2101
```

Mounting the newly copied root file system

First enable target 100 (local 2100) disk for Linux usage with the **chccwdev -e** command.

```
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
```

Determine which device name is the target 100, or local 2100, disk:

```
# lsdasd | grep 2100
0.0.2100   active      dasdi     94:32   ECKD  4096   2347MB    600840
```

The first target disk in this example is the device node /dev/dasdi. The first partition is the root file system, /dev/dasdi1. Mount it over the mount point /mnt and **cd** into it:

```
# mount /dev/dasdi1 /mnt
# cd /mnt
```

Observe that this appears to be a root file system:

```
# ls
bin   dev  home  lib64       media  opt   root  selinux  success  tmp  var
boot  etc  lib   lost+found  mnt    proc  sbin  srv      sys      usr
```

Modifying networking information on the target system

In this example, the only two pieces of networking information that are modified are the IP address (from ***9.60.18.222*** to ***9.60.18.224***) and the host name (from ***gpok222*** to ***gpok224***).

The host name is changed in the file /mnt/etc/HOSTNAME:

```
# cd etc
# vi HOSTNAME
gpok224.endicott.ibm.com
```

The IP address is changed in the file */mnt/etc/sysconfig/network/ifcfg-eth0*:

```
# cd sysconfig/network
# vi ifcfg-eth0
BOOTPROTO='static'
IPADDR='9.60.18.224/25'
BROADCAST='9.60.18.255'
STARTMODE='onboot'
NAME='OSA Express Network card (0.0.0600)'
```

Detach the target disks

Change the directory to /, use the **sync** command to flush the disks and the **umount** command to unmount the modified root file system:

```
# cd /
# sync
# umount /mnt
```

Remove the target disk from the cloner:

```
# chccwdev -d 2100
Setting device 0.0.2100 offline
```

```
Done
```

Detach the source and target minidisks using the CP **DETACH** command:

```
# vmcp det 1100
DASD 1100 DETACHED
# vmcp det 1101
DASD 1101 DETACHED
# vmcp det 2100
DASD 2100 DETACHED
# vmcp det 2101
DASD 2101 DETACHED
```

The newly copied and modified system disks have now been detached.

IPL the target system

You should now be ready to IPL the manually cloned system.

Start a 3270 session to LINUX01 and IPL from minidisk 100:

```
LOGON LINUX01
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0003 RDR,   NO PRT,   NO PUN
LOGON AT 15:34:17 EDT WEDNESDAY 09/01/10
z/VM V6.1.0    2010-08-30 14:21

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
zIPL v1.8.0-44.22.5 interactive boot menu

 0. default (SLES11_SP1)

 1. SLES11_SP1
 2. FailsafeV1
 3. ipl

Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'

Please choose (default will boot in 3 seconds):
Booting default (SLES11_SP1)...
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32.12-0.7-default (geeko@buildhost) (gcc version 4.3.4
Ÿgcc-4_
3-branch revision 152973¨ (SUSE Linux) ) #1 SMP 2010-05-20 11:14:20 +0200
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
...
```

Watch for error messages on the console. Your new system should come up cleanly using the modified IP address and host name.

Modify the SSH keys on the target system

Start an SSH session as root to the new system using the new IP address or host name.

The SSH keys that were copied are identical to those of the golden image. Manually create three new ones with the **ssh-keygen** command:

```
# cd /etc/ssh
# ssh-keygen -t rsa -N "" -q -f ssh_host_rsa_key
ssh_host_rsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t dsa -N "" -q -f ssh_host_dsa_key
ssh_host_dsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t rsa1 -N "" -q -f ssh_host_key
ssh_host_key already exists.
Overwrite (y/n)? y
```

Congratulations! You have now cloned a Linux system manually. You can look around the new system. It should be identical to the golden image except for the IP address and host name.

Next you learn how to do it automatically. You will use the LINUX01 user ID again. To clone, the target user ID must be logged off. You could shut the new system down cleanly, but because you will be cloning again, it does not matter. Go to the 3270 session and kill the LINUX01 user ID using the **LOGOFF** command:

```
==> #cp log
```

9.4 Cloning a virtual server automatically

Now that you have cloned a server manually and understand the steps better, you can use the clone.sh script to clone automatically.

Start an SSH session as root to the cloner. 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 two pairs of parameters:

from <user ID> The source Linux

to <user ID> The target Linux system

Both the source and target user IDs must be logged off. The script reads the parameter file on the LNXMAINT 192 disk (the cloner's 191 disk) to obtain information necessary to give the new Linux virtual server an identity. It calls CP **FLASHCOPY** using the **vmcp** command to try to copy the golden image's 100 and 103 disks. If **FLASHCOPY** fails, the script falls back to copying the disk using the Linux **dasdfmt** and **dd** commands. It then regenerates SSH keys and adds the server's public key to the cloner's known_hosts file.

Note: The clone.sh script changes only the IP address, the host name, and the domain. The Netmask is not changed.

The script then boots the new Linux using the **XAUTOLOG** command.

It should take less than a minute to clone with **FLASHCOPY** support and 10-20 minutes without it. Following is an example of cloning to the LINUX01 user ID with **FLASHCOPY** support. The output is divided into sections.

```
# clone.sh from s11s1gld to linux01
Checking that S11S1GLD exists and is not logged on ...
Invoking CP command: QUERY S11S1GLD
HCPCQU045E S11S1GLD not logged on
Error: non-zero CP response for command 'QUERY S11S1GLD': #45
Checking that LINUX01 exists and is not logged on ...
Invoking CP command: QUERY LINUX01
HCPCQU045E LINUX01 not logged on
Error: non-zero CP response for command 'QUERY LINUX01': #45
Setting device 0.0.0191 offline
Done
Setting device 0.0.0191 online
Done
S11S1GLD PARM-S11 D1 V         73          9          1  9/07/2010  9:17:09
LINUX01  PARM-S11 D1 V         73          9          1  9/01/2010 16:22:53

WARNING!!: Mindisks 100 and 101 will be copied to LINUX01
New host name will be:  gpok224.endicott.ibm.com
New IP address will be: 9.60.18.224
Other network data is retrieved from LINUX01 PARM-S11 on 191 disk
Are you sure you want to overwrite these disks (y/n): y
```

It is verified that the user IDs S11S1GLD and LINUX01 exist and are logged off. The 191 disk is deactivated and reactivated to pick up any changes that may have occurred. It is then verified that the two parameter files, S11S1GLD PARM-S11 and LINUX01 PARM-S11 exist. You are then asked to confirm that the disks on the target system can be written to. Answer yes with y.

The script copies the source 100 and 101 disks to the target user ID using the **FLASHCOPY** command. If **FLASHCOPY** is not supported or fails for some other reason, the script falls back to the **dasdfmt** and **dd** commands to copy the disks.

```
Linking source and target 100 disks ...
Invoking CP command: detach 1100
HCPDTV040E Device 1100 does not exist
Error: non-zero CP response for command 'DETACH 1100': #40
Invoking CP command: link S11S1GLD 100 1100 rr

Invoking CP command: detach 2100
HCPDTV040E Device 2100 does not exist
Error: non-zero CP response for command 'DETACH 2100': #40
Invoking CP command: link LINUX01 100 2100 mr

Copying 100 disks ...

FLASHCOPYing 1100 to 2100 ...
Invoking CP command: FLASHCOPY 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 3337 TO 2100 0 3337
Take 1100 Offline....
Device is already offline
Done
Invoking CP command: det 1100
DASD 1100 DETACHED
```

```
---------------------------------------
Linking source and target 101 disks ...
Invoking CP command: detach 1101
HCPDTV040E Device 1101 does not exist
Error: non-zero CP response for command 'DETACH 1101': #40
Invoking CP command: link S11S1GLD 101 1101 rr

Invoking CP command: detach 2101
HCPDTV040E Device 2101 does not exist
Error: non-zero CP response for command 'DETACH 2101': #40
Invoking CP command: link LINUX01 101 2101 mr

Copying 101 disks ...

FLASHCOPYing 1101 to 2101 ...
Invoking CP command: FLASHCOPY 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 3337 TO 2101 0 3337
Taking 1101 Offline...
Device is already offline
Done
Invoking CP command: det 1101
DASD 1101 DETACHED
Taking 2101 Offline...
Device is already offline
Done
Invoking CP command: det 2101
DASD 2101 DETACHED
Mounting newly cloned image over /mnt/targetLinux ...
Setting device 0.0.2100 online
Done
```

The script mounts the newly copied root file system over a mount point, `/mnt/targetLinux/`. The networking information is modified and the SSH keys are regenerated. Then the disk is detached.

```
Mounting /dev/dasdi1 over /mnt/targetLinux ...
Modifying cloned image under /mnt/targetLinux ...
Removing SSH keys
Removing 9.60.18.224 from known_hosts file
Setting device 0.0.2100 offline
Done
Invoking CP command: DETACH 2100
DASD 2100 DETACHED
```

In the final section, the target user ID is logged on using **`XAUTOLOG`**. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk `100`. The new system should be on the network in about 30-45 seconds.

```
Invoking CP command: XAUTOLOG LINUX01
Command accepted
Successfully cloned S11S1GLD to LINUX01
You should be able to ping 9.60.18.224 within one minute
```

Note: If the clone.sh script fails, you can 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 the LNXMAINT 192 disk.
- The target 100 and 101 minidisks are sized identically.

9.5 Creating three more virtual servers

So far you have installed Linux manually twice onto S11S1CLN and S11S1GLD. You have created the new user ID LINUX01 and cloned the golden image to it. Now it is time to clone three more times to have one system for each of the virtual servers described in the remaining chapters.

The following steps are involved:

- "Defining three more user IDs"
- "Testing logging on to a new user ID"
- "Creating three new parameter files"
- "Granting user IDs access to VSWITCH"
- "Testing logging on to a new user ID"

9.5.1 Defining three more user IDs

Define three more user IDs for Linux virtual servers in the USER DIRECT file named LINUX02 - LINUX04. You will need to use the DASD volumes you formatted in "Formatting DASD for minidisks" on page 166: two for each virtual server. You can repeat the definition of LINUX01 three times with the block copy **""3** prefix command. For example:

```
==> x user direct
====> /user linux01
====> -1
""3   *
02142 USER LINUX01 LNX4VM 256M 1G G
02143  INCLUDE LNXDFLT
02144  OPTION APPLMON
02145  MDISK 100 3390 3339 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
""     MDISK 101 3390 6677 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
```

This creates three more copies of the LINUX01 user definition. Modify them to have user IDs of LINUX02 - LINUX04, and give each correct DASD labels. In this example, two 3390-9s with labels UM63AA and UM63AB are used:

```
*
USER LINUX02 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3338 UM63AA MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 3339 3338 UM63AA MR LNX4VM LNX4VM LNX4VM
*
USER LINUX03 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 6677 3338 UM63AA MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 0001 3338 UM63AB MR LNX4VM LNX4VM LNX4VM
```

```
*
USER LINUX04 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 3339 3338 UM63AB MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 6677 3338 UM63AB MR LNX4VM LNX4VM LNX4VM
```

Go to the top of the file and find the definition for the user $ALLOC$. Add dummy definitions for cylinder 0 of each of the new volumes and save the changes:

```
====> top
====> /alloc
USER $ALLOC$  NOLOG
...
 MDISK A09 3390 000 001 UM63A9 R
 MDISK A0A 3390 000 001 UM63AA R
 MDISK A0B 3390 000 001 UM63AB R
...
====> file
```

Check for overlaps and the single gap. Use the **QUIT** command to exit the USER DISKMAP file:

```
==> diskmap user
==> x user diskmap
====> pre off
====> all /gap/|/overlap/
0         500         501    GAP
-------------------  6  line(s) not displayed --------------------
                                          0           0           1    GAP
-------------------  354  line(s) not displayed --------------------
====> quit
```

Bring the changes online with the **DIRECTXA USER** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 45 disk pages
```

You have now created three new user IDs that can be cloned to.

9.5.2 Creating three new parameter files

A new parameter must be created for each of the user IDs with the proper networking information. Perform the following steps:

- Log off of MAINT and log on to LNXMAINT.
- Copy the LINUX01 parameter file three times:

```
==> copy linux01 parm-s11 d linux02 = =
==> copy linux01 parm-s11 d linux03 = =
==> copy linux01 parm-s11 d linux04 = =
```

- Edit each of the three files replacing the appropriate network values. For example, in the LINUX02 PARM-S11 file, only the TCP/IP address and DNS name need to be modified because all other network and other values are the same:

```
==> x linux02 parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
```

```
HostIP=9.60.18.225 Hostname=gpok225.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.223/nfs/sles11sp1/SLES-11-SP1-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
====> file
==> x linux03 parm-s11 d
...
```

You should now have three new parameter files.

9.5.3 Granting user IDs access to VSWITCH

Modify the PROFILE EXEC on AUTOLOG1 191 to grant access to the VSWITCH for the three new user IDs and add **XAUTOLOG** commands so they will boot when the z/VM system IPLs. Perform the following steps:

- 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 file:

```
==> x profile exec f
...
/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant s11s1cln'
'cp set vswitch vsw1 grant s11s1gld'
'cp set vswitch vsw1 grant linux01'
'cp set vswitch vsw1 grant linux02'
'cp set vswitch vsw1 grant linux03'
'cp set vswitch vsw1 grant linux04'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog s11s1cln'
'cp xautolog linux01'
'cp xautolog linux02'
'cp xautolog linux03'
'cp xautolog linux04'
'cp logoff'                          /* logoff when done */
====> file
```

- It is easiest to grant access to the new user IDs for the current z/VM session with the **SET VSWITCH** command:

```
==> set vswitch vsw1 grant linux02
Command complete
==> set vswitch vsw1 grant linux03
Command complete
==> set vswitch vsw1 grant linux04
Command complete
```

- Verify that the user IDs have access with the **QUERY VSWITCH ACCESSLIST** command:

```
==> q vswitch vsw1 acc
```

```
...
    Authorized userids:
      LINUX01  LINUX02  LINUX03  LINUX04  SYSTEM   S11S1CLN
      S11S1GLD
  RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
  RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1  BACKUP
```

You have now defined three new user IDs suitable for cloning the golden image to.

9.5.4 Testing logging on to a new user ID

You should now be able to log on to a new user ID and verify the integrity of the definitions. Perform the following steps:

- Log on to LINUX02; you should first notice that a NIC is created as well as two VDISKs:

```
LOGON LINUX02
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 07:55:27 EDT THURSDAY 09/02/10
z/VM V6.1.0    2010-08-30 14:21

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

- If you forgot to grant access to the VSWITCH, you will see an error message. Verify that you have OSA devices at addresses 600-602, and read-write DASD devices at addresses 100-102:

```
==> q osa
OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
     0600 DEVTYPE OSA         CHPID 00 OSD
...
==> q da
DASD 0100 3390 UM63A9 R/W       3338 CYL ON DASD  63A9 SUBCHANNEL = 0000
DASD 0101 3390 UM63AA R/W       3338 CYL ON DASD  63AA SUBCHANNEL = 0001
DASD 0190 3390 610RES R/O        107 CYL ON DASD  6280 SUBCHANNEL = 0009
DASD 0191 3390 UM6289 R/O        300 CYL ON DASD  6289 SUBCHANNEL = 000C
DASD 019D 3390 UV6283 R/O        146 CYL ON DASD  6283 SUBCHANNEL = 000A
DASD 019E 3390 UV6283 R/O        250 CYL ON DASD  6283 SUBCHANNEL = 000B
DASD 0300 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
DASD 0301 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000F
DASD 0592 3390 UV6284 R/O         70 CYL ON DASD  6284 SUBCHANNEL = 000D
```

- Log off of LINUX02.

Congratulations, you have cloned one Linux virtual server and defined three more user IDs that should now be ready for cloning to. You will clone to some of these user IDs in the chapters that follow.

9.6 Reviewing system status

Step back now and view your system from a DASD point of view. An example is shown in Figure 9-1. The DASD used by the examples in this book are twelve 3390-3 (eight for the /VM system and four for `LNXMAINT` and the majority of `S11S1CLN`) and four 3390-9s for the golden image and the remaining Linux virtual servers.

The possible role of administrators and users is suggested on the right side of Figure 9-1.

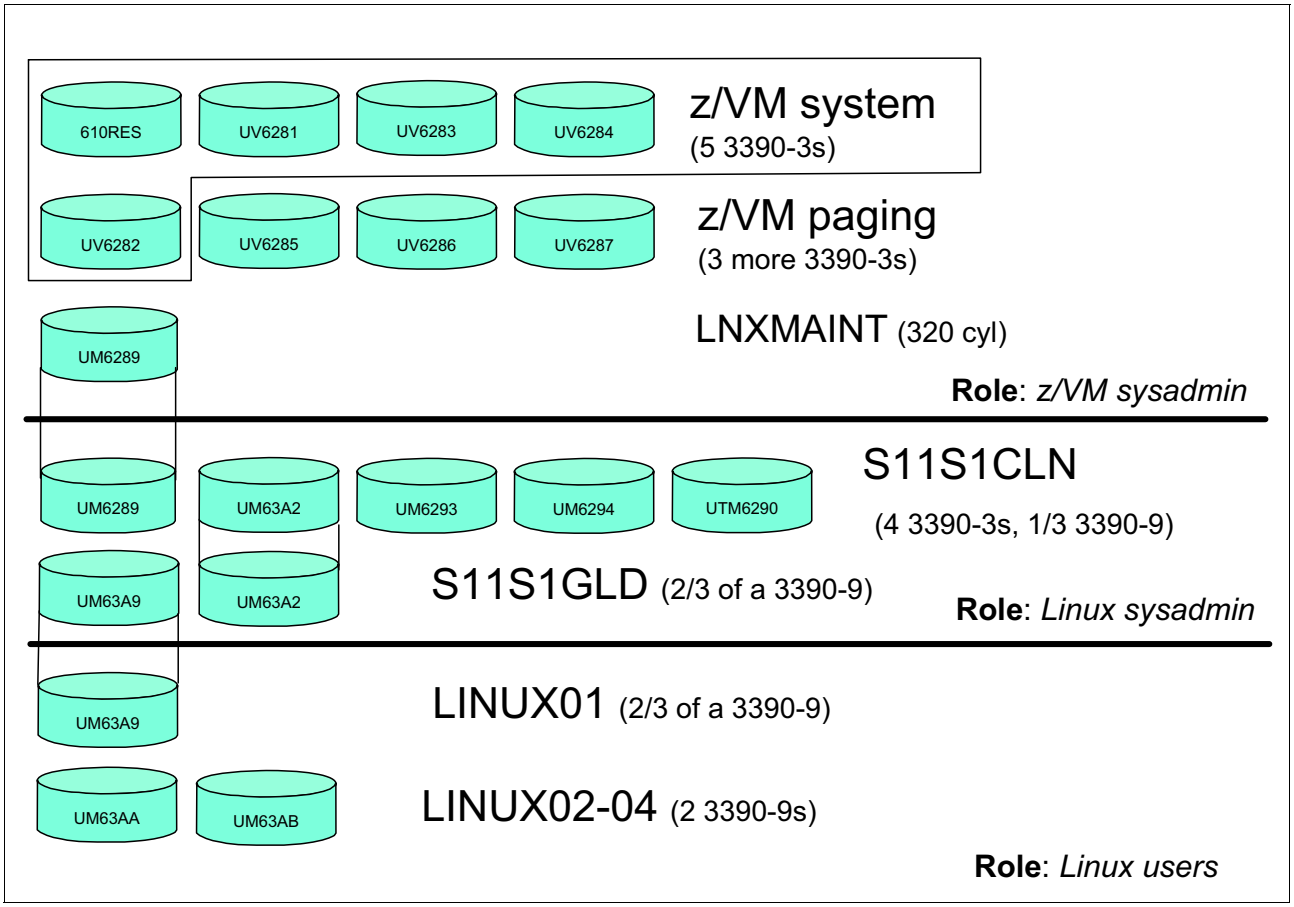


Figure 9-1 Linux virtual server system - DASD view and role view

These disk allocations can also be viewed with the **`QUERY DASD`** command (from `MAINT`):

```
==> q da
DASD 6280 CP OWNED  610RES   70
DASD 6281 CP OWNED  UV6281   1
DASD 6282 CP OWNED  UV6282   0
DASD 6283 CP OWNED  UV6283   87
DASD 6284 CP OWNED  UV6284   27
DASD 6285 CP OWNED  UP6285   0
DASD 6286 CP OWNED  UP6286   0
DASD 6287 CP OWNED  UP6287   0
DASD 6289 CP SYSTEM UM6289   3
DASD 6290 CP SYSTEM UM6290   1
DASD 6293 CP SYSTEM UM6293   1
DASD 6294 CP SYSTEM UM6294   1
DASD 63A2 CP SYSTEM UM63A2   1
```

```
DASD 63A9 CP SYSTEM UM63A9   2
DASD 63AA CP SYSTEM UM63AA   0
DASD 63AB CP SYSTEM UM63AB   0
```

The next step will be to start cloning virtual servers for different uses to create *appliances*.

10

Cloning open source virtual servers

This chapter describes how to clone and customize the following Linux virtual servers:

- “Creating a virtual web server”
- “Creating a virtual LDAP server”
- “Creating a virtual file and print server”
- “Creating a virtual application development server”

These Linux virtual servers can be thought of as *virtual appliances* once they have been cloned and *personalities* have been added to them.

10.1 Creating a virtual web server

The example in this section uses the LINUX01 user ID to create a virtual web server. You should have a vanilla virtual server cloned to the user ID LINUX01 as described in Chapter 9, “Cloning SLES 11 SP1” on page 165.

10.1.1 Installing Apache RPMs

This section describes how to install the Apache web server RPMs.

1. **SSH** into the IP address of the new LINUX01 server.
2. Install the following Apache RPMs using the **zypper install** command:

```
# zypper install apache2-prefork apache2 apache2-doc apache2-example-pages
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW packages are going to be installed:
  apache2 apache2-doc apache2-example-pages apache2-prefork apache2-utils
  libapr-util1 libapr1

7 new packages to install.
Overall download size: 3.0 MiB. After the operation, additional 13.8 MiB will
be used.
Continue? [y/n/?] (y): y
...
Installing: apache2-prefork-2.2.10-2.24.5 [done]
```

3. Confirm that the RPMs have been added using the **rpm -qa** command:

```
# rpm -qa | grep apache
apache2-utils-2.2.10-2.24.5
apache2-example-pages-2.2.10-2.24.5
apache2-2.2.10-2.24.5
apache2-doc-2.2.10-2.24.5
apache2-prefork-2.2.10-2.24.5
```

10.1.2 Testing Apache

Start the Apache web server to verify that it is installed successfully.

1. Start the Apache server using the **service** command and set it to start at boot time using the **chkconfig** command:

```
# service apache2 start
Starting httpd2 (prefork)                                          done
# chkconfig apache2 on
```

2. To verify that Apache is installed correctly, after it has been started, point a web browser to the server and see the Apache test page. In your web browser, put in the host name or IP address of your web server as the URL. In this example the virtual server running on LINUX01 has a DNS name of gpok224.endicott.ibm.com:

   http://***gpok224.endicott.ibm.com***

3. You should see a test page with two words: It works!

If you get an error starting Apache, look in the log file `/var/log/apache2/error-log` for clues. If Apache started successfully but you cannot reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

10.1.3 Populating your website

You can begin to put your web pages in the directory `/srv/www/htdocs/`, which is the default web root. For security and customization purposes, you might want to change the default web root to point to another directory. The easiest way to do this is to copy `/etc/apache2/default-server.conf` to your own configuration file, that is, `/etc/apache2/my-server.conf`. Make the changes in `/etc/apache2/my-server.conf`, and then edit `/etc/apache2/httpd.conf` to use `my-server.conf`.

10.1.4 Apache resources

The following web sites contain additional information on 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 using 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 will clone Linux and configure a new virtual LDAP server. Then you will go back to the virtual web server you just created and point it to the new LDAP server.

Then you may want to configure the golden image so that it is pointing to this virtual server. If you do so, all Linux images that are cloned will be able to use this virtual LDAP server.

The steps in this section are as follow:

- “Cloning a Linux”
- “Configuring the LDAP server”
- “Adding an LDAP user”
- “Setting another virtual server to use the LDAP server”

10.2.1 Cloning a Linux

From a root session on the cloner, clone from the golden image (user ID `S11S1GLD`) to `LINUX02` using the clone.sh script:

```
# clone.sh from s11s1gld to linux02
# clone.sh from s11s1gld to linux02
Checking that S11S1GLD exists and is not logged on ...
Invoking CP command: QUERY S11S1GLD
HCPCQU045E S11S1GLD not logged on
Error: non-zero CP response for command 'QUERY S11S1GLD': #45
Checking that LINUX02 exists and is not logged on ...
Invoking CP command: QUERY LINUX02
```

```
HCPCQU045E LINUXO2 not logged on
Error: non-zero CP response for command 'QUERY LINUXO2': #45
Setting device 0.0.0191 offline
Done
Setting device 0.0.0191 online
Done
the program '/bin/bash' called 'udevsettle', it should use 'udevadm settle
<options>', this will stop working in a future release
S11S1GLD PARM-S11 D1 V         73          9          1  9/01/2010 14:00:27
LINUXO2  PARM-S11 D1 V         73          9          1  9/01/2010 17:18:57

WARNING!!: Mindisks 100 and 101 will be copied to LINUXO2
New host name will be:  gpok225.endicott.ibm.com
New IP address will be: 9.60.18.225
Other network data is retrieved from LINUXO2 PARM-S11 on 191 disk
Are you sure you want to overwrite these disks (y/n): y
...
```

The clone.sh script should create a new virtual server.

10.2.2 Configuring the LDAP server

To configure the OpenLDAP server, the **yast** tool is recommended.

- Start an SSH session as `root` to the new server.
- Invoke the **yast** command. The `YaST Control Center` should appear.
- Use the down arrow key to move to Network Services on the left side. Use the Tab or right arrow key to move to the right side and select **LDAP Server**. Press Enter.

```
# yast
```

```
+--------------------------------------------------------------------------+
|                          YaST2 Control Center                            |
+--------------------------------------------------------------------------+

+-------------------+ +-----------------------------------------------+
|Software           | |Hostnames                                      |
|Hardware           | |Kerberos Client                                |
|System             | |Kerberos Server                                +
|Network Devices    | |LDAP Browser                                   |
|Network Services   | |LDAP Client                                    |
|Security and Users | |LDAP Server                                    +
|Support            | |Mail Transfer Agent                            |
|Miscellaneous      | |NFS Client                                     |
|                   | |NFS Server                                     |
|                   | |NIS Client                                     |
|                   | |NIS Server                                     |
+-------------------+ +-----------------------------------------------+
```

- You should see a pop-up window with the following message. Press Enter to accept the default of **Install** and some RPMs will be installed:

  ```
  These packages need to be installed: openldap2
  ```

- The LDAP Server Configuration panel will appear. Accept the default of **Yes** by using the tab key to **Next** and pressing Enter.

```
                    General Settings
  +Start LDAP Server-------------------------------+
  |(x) Yes                                         |
  |( ) No                                          |
  |[ ] Register at an SLP Daemon                   |
  +------------------------------------------------+

  +Firewall Settings-------------------------------+
  |[ ] Open Port in Firewall  [Firewall Details...]|
  |Firewall is disabled                            |
  +------------------------------------------------+
```

- A second LDAP Server Configuration panel will appear. Accept the default of **standalone server** by using the tab key to **Next** and pressing Enter.

```
                          Please select Server type

   (x) This will be a standalone server

   ( ) This server can act as a master server in a replication setup

   ( ) This will be a replica (slave) server.
       All data including configuration will replicated from a remote server.
```

- In the TLS settings panel, accept all defaults of using TLS and pressing **Next**. This will ensure that LDAP communications are encrypted.
- The Basic Database Settings panel will appear. The `Base DN` (in this example `dc=endicott,dc=ibm,dc=com`) should be correct for your DNS domain. Set the LDAP administrator password (twice) and press **Next**.

```
New Database
                          Basic Database Settings
         Database Type
         hdb#########â
         Base DN
         dc=endicott,dc=ibm,dc=com################################

         Administrator DN
         cn=Administrator###################### [x] Append Base DN

         LDAP Administrator Password
         ******##################################################
         Validate Password
         ******##################################################

         Database Directory
         /var/lib/ldap################################# [Browse...]
         [x] Use this database as the default for OpenLDAP clients
```

- The LDAP Server Configuration Summary panel should appear now, this time with one database listed identified by the Base DN (`dc=endicott,dc=ibm,dc=com` in this example). Select **Finish** by pressing Enter.

```
LDAP Server Configuration Summary
+------------------------------------------------------------------------+
|Startup Configuration                                                   |
|                                                                        |
|Start LDAP Server: Yes                                                  |
|                                                                        |
|Register at SLP Service: No                                             |
|                                                                        |
|Create initial Database with the following Parameters                   |
|                                                                        |
|Database Suffix: dc=endicott,dc=ibm,dc=com                              |
|                                                                        |
|Administrator DN: cn=Administrator,dc=endicott,dc=ibm,dc=com            |
|                                                                        |
+------------------------------------------------------------------------+
```

- The panel Saving LDAP Server Configuration should appear. The database will be created and the LDAP server configured.
- You should be returned to the YaST Control Center. Move the cursor to **Quit** and press Enter.
- Verify that the LDAP server is running with the **service** command and that it is set to start in runlevels 3 and 5 with the **chkconfig** command:

```
# service ldap status
Checking for service ldap:                                          running
# chkconfig --list ldap
ldap                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

You have now cloned a new virtual server and configured it to run OpenLDAP.

10.2.3 Adding an LDAP user

When the golden image was installed, we recommended that a non-root user ID be added. In this example, it was `mikemac`.

- Verify that this user exists with the **id** command and see that there is an entry in the `/etc/passwd` file with the **grep** command:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=16(dialout),33(video),100(users)
# grep mikemac /etc/passwd
mikemac:x:1000:100::/home/mikemac:/bin/bash
```

- Delete this local user using the **userdel** command so it can be added to LDAP later.

```
# userdel mikemac
no crontab for mikemac
# id mikemac
id: mikemac: No such user
```

- An LDIF (LDAP Interchange Format) file is created to add an organizational unit named `People` and a user ID named `mikemac`. Create a similar file for your system's values:

```
# cd /var/lib/ldap
# vi initial.ldif // create the input file ...
dn: ou=People,dc=endicott,dc=ibm,dc=com
ou: People
```

```
objectClass: top
objectClass: organizationalUnit

dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac
```

- Add the contents of the LDIF file to the LDAP server with the **ldapadd** command (the line wraps, but it is one command):

```
# ldapadd -x -h localhost -D "cn=Administrator,dc=endicott,dc=ibm,dc=com" -f
initial.ldif -W
Enter LDAP Password:
adding new entry "ou=People,dc=endicott,dc=ibm,dc=com"

adding new entry "uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com"
```

- Search for the new user ID just added with the **ldapsearch** command:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, endicott.ibm.com
dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

- This shows that the user ID exists in the LDAP database. Now you may want to set the password with the **ldappasswd** command. You will need to provide a new password for the new user and you will also need to provide the LDAP administrator password.

```
# ldappasswd -x -D "cn=Administrator,dc=endicott,dc=ibm,dc=com" -W -S
"uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com"
New password:
Re-enter new password:
Enter LDAP Password:
Result: Success (0)
```

You have now deleted a local user, added a new LDAP user using an LDIF file, and set the new LDAP user's password.

10.2.4 Setting another virtual server to use the LDAP server

Now that you have a virtual LDAP server, you may want to point another virtual server to it so that you will have a centralized user database. If you have been following along in this book you should have created a web server running on the LINUX01 user ID. To point it to an LDAP server is fairly easy. In this section you will perform the following steps:

- "Testing that the LDAP client is not working"
- "Using YaST to modify the LDAP authentication client"
- "Testing the LDAP client"

Testing that the LDAP client is not working

Before you start, try a couple of commands to show that LDAP is *not* working. Get an SSH session to the virtual web server running on the user ID LINUX01.

Search for the LDAP user ID that you added earlier to the virtual LDAP server. In this example it is mikemac.

```
# ldapsearch -x uid=mikemac
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

The **ldapsearch** command cannot resolve the LDAP user because it cannot contact the LDAP server.

Delete the non-root user (mikemac in this example) from the local file system with the **userdel** command:

```
# userdel mikemac
no crontab for mikemac
```

Using YaST to modify the LDAP authentication client

The **yast** system administration interface can be used to configure the LDAP authentication client.

1. Invoke the **yast** command. The YaST Control Center should appear.
2. Select **Network Services** on the left side and **LDAP Client** on the right. Press Enter:

```
# yast
  +------------------------------------------------------------------------+
  ¦                          YaST Control Center                           ¦
  +------------------------------------------------------------------------+

  +-------------------+ +---------------------------------------------+
```

```
¦Software            ¦ ¦DNS and Hostname                                ¦
¦Hardware            ¦ ¦Hostnames                                       ¦
¦System              ¦ ¦Kerberos Client                                 ¦
¦Network Devices     ¦ ¦LDAP Browser                                    ¦
¦Network Services    ¦ ¦LDAP Client                                     ¦
¦Security and Users  ¦ ¦LDAP Server                                     ¦
¦Miscellaneous       ¦ ¦Mail Transfer Agent                             ¦
¦                    ¦ ¦NFS Client                                      ¦
...
```

3. On the LDAP Client Configuration panel, perform the following steps:
   a. Use the Tab key to move to **Use LDAP** and press the space bar to select that choice.
   b. Move to the Addresses of LDAP Servers field and enter the IP address (or DNS name) of your LDAP server. You can either enter LDAP base DN manually or press **Fetch DN** and then **OK** in result window. This way you can make sure that the LDAP server is accessible.
   c. Deselect LDAP TLS/SSL

```
+User Authentication--------------------------+
¦                                             ¦
¦ ( ) Do Not Use LDAP                         ¦
¦ (x) Use LDAP                                ¦
¦ ( ) Use LDAP but Disable Logins             ¦
¦                                             ¦
+---------------------------------------------+
+LDAP Client----------------------------------+
¦                                             ¦
¦ Addresses of LDAP Servers                   ¦
¦ 9.60.18.225_____________________[Find]      ¦
¦ LDAP Base DN                                ¦
¦ dc=endicott,dc=ibm,dc=com_______[Fetch DN]  ¦
¦ [ ] LDAP TLS/SSL                            ¦
¦ [ ] LDAP Version 2                          ¦
+---------------------------------------------+
```

4. Use the Tab key to move to **OK** and press Enter. You should get the following prompt. Press Enter to continue:

```
These packages need to be installed:
 pam_ldap
 nss_ldap
 pam_ldap-32bit
 nss_ldap-32bit
```

5. Accept **OK** if you get a warning window. Your changes will be saved.
6. At the main YaST2 Control Center, press **Quit** on the main window to quit YaST.

Your web server virtual Linux should now also be using OpenLDAP for user and group ID resolution and authentication.

Testing the LDAP client

Try the **id** command against the new LDAP user:

```
# id mikemac
uid=501(mikemac) gid=100(users) groups=100(users)
```

Note that the UID is 501 in this example (from the LDIF file), not the value 1000 (from the Linux installation).

Try the **ldapsearch** command again:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <dc=endicott,dc=ibm,dc=com> (default) with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, endicott.ibm.com
dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

You should also be able to start an SSH session to the virtual web server using the LDAP user.

You may also want to set the golden image to authenticate with the LDAP server. In this fashion, all virtual servers cloned after that will be able to utilize a centralized authentication server.

10.3 Creating a virtual file and print server

Samba allows Windows clients to map Linux file systems as shared drives. Samba can also act as a middleman 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 follows:

- “Cloning a Linux virtual server”
- “Installing necessary RPMs”
- “Configuring the Samba configuration file”
- “Adding a Samba user”

- “Starting Samba at boot time”
- “Testing your changes”

10.3.1 Cloning a Linux virtual server

Start an SSH session to the cloner as `root`, clone a basic virtual server. In this example the user ID `LINUX03` is used.

```
# clone.sh from s11s1gld to linux03
...
```

SSH in to the new virtual server.

10.3.2 Installing necessary RPMs

Add the following RPMs with the **`zypper install`** command:

```
# zypper install samba yast2-samba-server samba-doc samba-winbind cups
cups-drivers
...
```

You will see a number of YaST windows flash by as the RPMs are added to the system.

Confirm that the RPMs were added:

```
# rpm -qa | egrep "samba|cups"
cups-libs-1.3.9-8.30.1
yast2-samba-server-2.17.11-0.2.72
samba-doc-3.4.3-1.17.2
samba-client-32bit-3.4.3-1.17.2
samba-3.4.3-1.17.2
cups-drivers-1.3.9-2.31
cups-libs-32bit-1.3.9-8.30.1
yast2-samba-client-2.17.17-0.2.23
samba-client-3.4.3-1.17.2
samba-winbind-32bit-3.4.3-1.17.2
samba-32bit-3.4.3-1.17.2
cups-1.3.9-8.30.1
cups-client-1.3.9-8.30.1
samba-winbind-3.4.3-1.17.2
```

The Samba and CUPS RPMs are now installed.

10.3.3 Configuring the Samba configuration file

The one configuration file for Samba is `/etc/samba/smb.conf`. It is easy to add an SMB share that will be made available by the Samba server. A good test directory is `/usr/share/doc/` because it has much good Linux documentation. The following example creates a file *share* named `sharedoc`:

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf    // add three lines at the bottom of the file:
...
[sharedoc]
        comment = SLES 11 SP1 on System z documentation
```

```
path = /usr/share/doc/
```

This will cause an SMB share named **sharedoc** consisting of the contents of `/usr/share/doc` to be created when Samba is started.

10.3.4 Adding a Samba user

The default method that Samba uses to determine a user's credentials is to look in the `/etc/samba/smbpasswd` file. That user must first exist in the Linux file system (`/etc/passwd`, `/etc/shadow`, and so on).

The following example shows adding the user `mikemac` to the `smbpasswd` file. In this example the user exists in the local file system. Perform the following steps:

1. Use the **id** command to verify the local user:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=16(dialout),33(video),100(users)
```

2. To create a new Samba user, the **smbpasswd -a** command is used. If there are policy error messages, they can be ignored.

```
# smbpasswd -a mikemac
New SMB password:
Retype new SMB password:
account_policy_get: tdb_fetch_uint32 failed for field 1 (min password length),
returning 0
...
Added user mikemac.
```

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 10.2.4, "Setting another virtual server to use the LDAP server" on page 190, because the Samba schema must first be added to LDAP.

10.3.5 Starting Samba at boot time

Samba consists of two daemons, nmbd and smbd. They can be started for the current session with the **service** command for both the nmb and smb daemons:

```
# service nmb start
Starting Samba NMB daemon                                          done
# service smb start
Starting Samba SMB daemon                                          done
```

The following **chkconfig** commands will set these daemons to start at boot time:

```
# chkconfig nmb on
# chkconfig smb on
```

Samba should now be running and configured to start at boot time.

10.3.6 Testing your changes

You can verify that the Samba daemons are running with the **service** command:

```
# service nmb status
```

```
Checking for Samba NMB daemon                              running
# service smb status
Checking for Samba SMB daemon                              running
```

You can test getting a Samba share from a Windows desktop:

1. Go to any Windows Explorer window (such as My Computer) and select **Tools** → **Map Network Drive**.
2. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 10-1. In this example the UNC is ***9.60.18.226*\sharedoc**.
3. You may need to click **Connect using different user name**, if the sample user ID and password are different on your desktop computer from the values you set on the Samba server.
4. Click **Finish**. If all the steps were correct, you should see the files in a new Explorer window as shown in the bottom right corner of Figure 10-1.

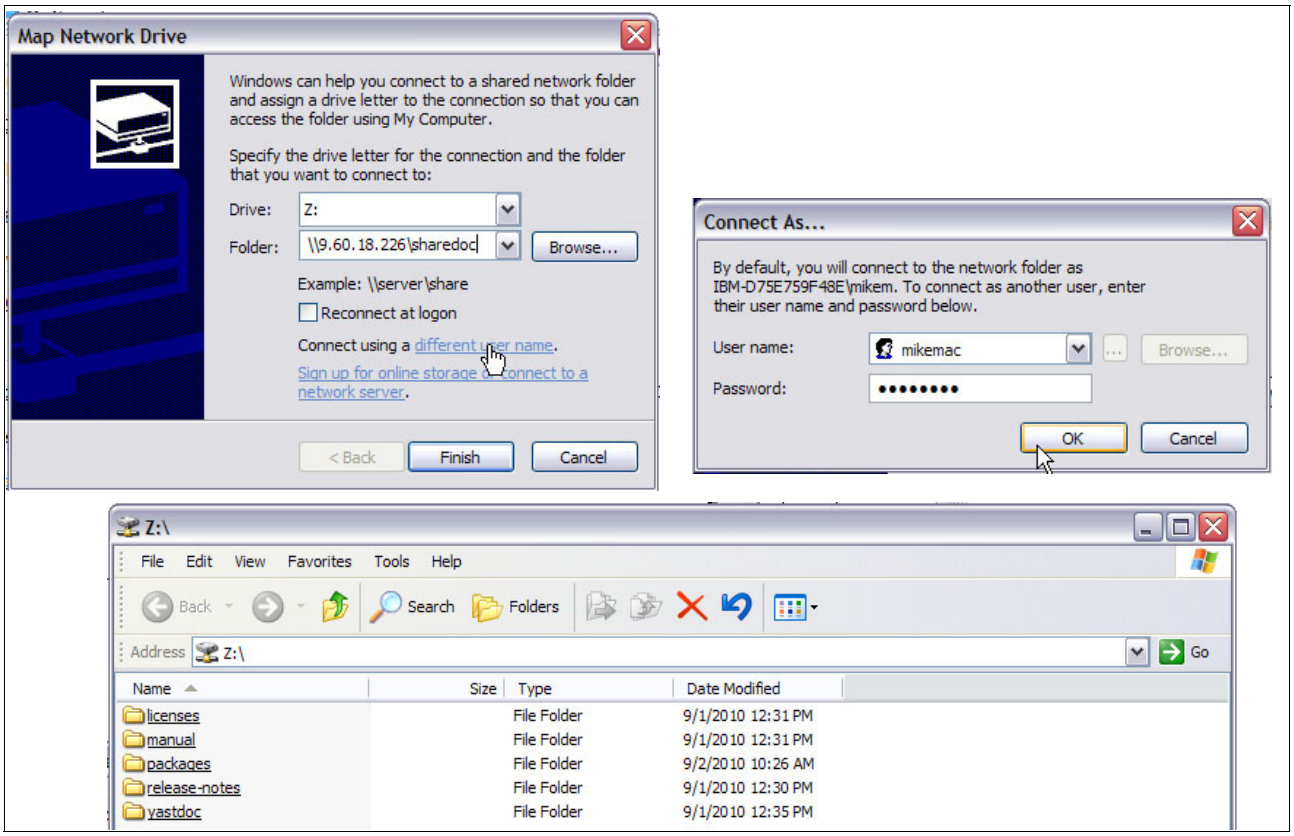


Figure 10-1 Mapping a network drive to a Samba share

You should now have Samba configured and running with one new share available.

If you prefer a command line, you can map a network drive with the DOS **NET USE** command:

```
c:\> net use * \\9.60.18.226\sharedoc
The password is invalid for \\9.60.18.226\sharedoc.

Enter the user name for '9.60.18.226': mikemac
Enter the password for 9.60.18.226:
Drive Z: is now connected to \\9.60.18.226\sharedoc.
```

```
The command completed successfully.
```

Another useful DOS command to interrogate the Samba server is the **NET VIEW** command. For example:

```
The command completed successfully.
c:\> net view \\9.60.18.226
Shared resources at \\9.60.18.226

Samba 3.4.3-1.17.2-2359-SUSE-CODE11

Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
groups      Disk           All groups
mikemac     Disk           Home Directories
profiles    Disk           Network Profiles Service
sharedoc    Disk  Z:       SLES 11 SP1 on System z documentation
users       Disk           All users
The command completed successfully.
```

10.3.7 Configuring printing

Configuring printing is more complex and is beyond the scope of this section. For details see the IBM Redpaper *Printing with Linux on zSeries® Using CUPS and Samba*, REDP-3864 at:

http://www.redbooks.ibm.com/abstracts/redp3864.html?Open

10.4 Creating a virtual application development server

Most Linux distributions come with a robust 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, and memory management.
- Application development (C/C++ and Java) for software to be used on Linux.
- Web development for applications to be run on the web, such as stock trade applications or e-mail applications.

The development languages used in implementation range from scripting languages such as Python or Tcl, to compiled languages such as C/C++ and Java. There is software available on Linux 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. At 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

http://www.sourceforge.net

10.4.1 Cloning a Linux virtual server

From the cloner clone a basic virtual server to LINUX04.

```
# clone.sh from s11s1gld to linux04
...
```

Start an SSH session as root to the new virtual server.

You can install *all* the application development tools described in this section with the following command:

```
# zypper install python perl tcl php gcc gdb make java-1_6_0-ibm
...
17 new packages to install.
Overall download size: 65.1 MiB. After the operation, additional 151.2 MiB will
be used.
Continue? [y/n/?] (y): y
...
```

If you want to install only certain tools, each specific RPM or group of RPMs is described in the sections that follow.

10.4.2 Scripting languages

Scripts are good for quickly automating a process or writing your own commands. They are also used for 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

  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:

  ```
  # zypper install python
  ...
  ```

- Perl

  Practical Extraction and Report Language. Perl is optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is also 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:

  ```
  # zypper install perl
  ...
  ```

- Tcl

  Tcl, the Tool Command Language, is a very simple programming language. It 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 that 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

```
# zypper install tcl
...
```

- PHP

  PHP (recursive acronym for "PHP: Hypertext Preprocessor") 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:

```
# zypper install php
...
```

10.4.3 C/C++ development tools

Most Linux distributions come with the C/C++ compiler, `gcc`. This is also known as the *GNU compiler collection* because it can compile other languages such as Fortran, but it is most frequently used to compile C and C++ code. In the minimal SLES10 installation, none of the development packages are installed. In order to use `gcc`, you must install it using yast:

```
# zypper install gcc
...
# rpm -qa | grep gcc
libgcc43-4.3.4_20091019-0.7.35
gcc43-4.3.4_20091019-0.7.35
gcc-4.3-62.198
libgcc43-32bit-4.3.4_20091019-0.7.35
```

`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 require preprocessing, compilation, and assembly first to create object files, then linking to combine 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:

```
# zypper install gdb
```

There is a good tutorial on getting started with gdb at:

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. To install make, issue the command:

```
# zypper install make
```

10.4.4 Java development tools

SLES 11 comes with IBM Java Standard Development Kit (SDK) which is needed if you want to develop Java applications. You need a Java Runtime Environment (JRE) if you only want to run Java applications. The following command installs both:

```
# zypper install java-1_6_0-ibm
```

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 is a great guide to get you started with either tool:

http://www.onlamp.com/pub/a/onlamp/2004/11/18/gnumake_3e.html

10.4.5 Additional resources

The following web sites are resources for additional information:

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

Java

http://www.oracle.com/technetwork/java/index.html
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/

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 DASD

The following process describes how to add additional DASD to a Linux guest. The overall steps are:

- "Adding minidisks to LINUX02"
- "Making the new minidisks available"
- "Formatting and partitioning the minidisks"
- "Creating a logical volume and file system"
- "Updating the file system table"

11.1.1 Adding minidisks to LINUX02

Following are the high-level steps to add two new 3390-3-sized minidisks to `LINUX02`:

- Determine the volume or volumes that will be added. In this example, a 3390-3 at real device address 6339 is added. Its space is split in half.
- Add minidisk statements to define minidisks. In this example, two minidisks at virtual addresses `102` and `103` are defined of size 1669 cylinders to the `LINUX02` user ID.
- Create the `USER DISKMAP` file to verify the disk layout.
- Bring the changes online with the **`DIRECTXA`** command.
- Shut down the Linux system.
- Log off the user ID.
- Log back on to it and IPL Linux.

Following is the updated directory entry:

```
USER LINUX02 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3338 UM63AA MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 3339 3338 UM63AA MR LNX4VM LNX4VM LNX4VM
 MDISK 102 3390 0001 1669 UM6339 MR LNX4VM LNX4VM LNX4VM
 MDISK 103 3390 1670 1669 UM6339 MR LNX4VM LNX4VM LNX4VM
```

11.1.2 Making the new minidisks available

When your system comes back up, start an SSH session to it. Use the **`lsdasd`** command to verify that the new minidisks are not seen yet (because there is `dasd=100-102` in `/etc/zipl.conf`):

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2347MB    600840
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0301   active      dasdd     94:12   FBA   512    512MB     1048576
```

Bring disks `102` and `103` online with the **`dasd_configure`** command and verify that they are available with the **`lsdasd`** command:

```
# dasd_configure 0.0.0102 1
Configuring device 0.0.0102
```

```
Setting device online
# dasd_configure 0.0.0103 1
Configuring device 0.0.0103
Setting device online
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2347MB    600840
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0301   active      dasdd     94:12   FBA   512    512MB     1048576
0.0.0102   active      dasde     94:16   ECKD  4096   1173MB    300420
0.0.0103   active      dasdf     94:20   ECKD  4096   1173MB    300420
```

You should be ready to format the new minidisks.

11.1.3 Formatting and partitioning the minidisks

You could format the minidisks sequentially, but you can also use the following bash **for** loop to put two dasdfmt jobs in the background so as to format both minidisks in parallel:

```
# for i in e f
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 2713
[2] 2714
```

When the jobs are finished, you should see messages similar to the following:

```
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok
```

Now use the **fdasd** command with the **-a** flag to create a single partition from each minidisk:

```
# fdasd -a /dev/dasde
reading volume label ..: VOL1
reading vtoc ..........: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
# fdasd -a /dev/dasdf
reading volume label ..: VOL1
reading vtoc ..........: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The minidisks are now ready for you to use.

If you are creating a new logical volume, see 11.2.1, “Creating a logical volume and file system” on page 204. If you are extending an existing logical volume, skip ahead to 11.3, “Extending an existing logical volume” on page 208.

11.2 Adding a logical volume

There are times when you require more disk space than a single direct access storage device (DASD) volume provides. For example, if you want to have a shared `/home/` directory, you want it to be of sufficient size. When this is the case, you can use the Logical Volume Manager (LVM) to combine multiple DASD volumes into one logical volume.

The following process describes how to create a logical volume with additional DASD on a Linux guest. The overall steps in adding a logical volume are:

- “Adding DASD” on page 202
- “Creating a logical volume and file system”
- “Updating the file system table”

11.2.1 Creating a logical volume and file system

The overall steps involved in creating a logical volume are:

- Create physical volumes from the two partitions.
- Create a single volume group.
- Create a single logical volume.
- Make a file system from the logical volume.

Figure 11-1 on page 204 shows a block diagram of the logical volume manager reflecting this example.

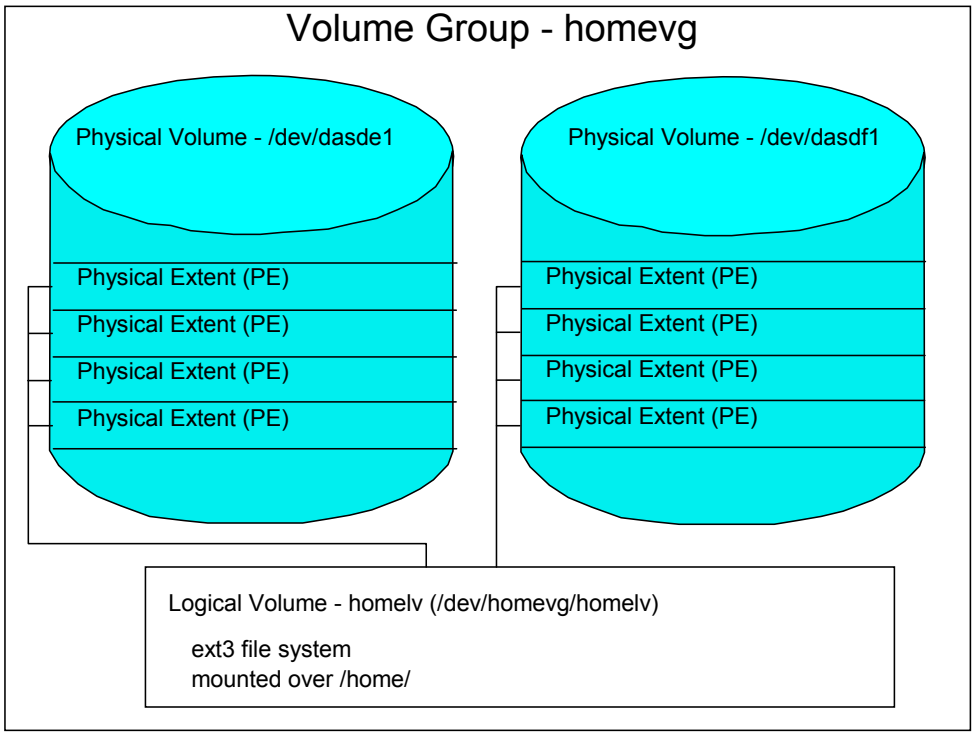


Figure 11-1 LVM block diagram

Creating physical volumes from the two DASD

The **pvcreate** command initializes partitions for use by LVM. Initialize the two new DASD partitions.

```
# pvcreate /dev/dasde1 /dev/dasdf1
  No physical volume label read from /dev/dasde1
  Physical volume "/dev/dasde1" successfully created
  No physical volume label read from /dev/dasdf1
  Physical volume "/dev/dasdf1" successfully created
```

Verify that the physical volumes were created with the **pvdisplay** command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
"/dev/dasde1" is a new physical volume of "1.15 GB"
  --- NEW Physical volume ---
  PV Name               /dev/dasde1
  VG Name
  PV Size               1.15 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               VsBCkw-peLr-45CP-VROC-dXOk-Jc9B-kIhOOl

  "/dev/dasdf1" is a new physical volume of "1.15 GB"
  --- NEW Physical volume ---
  PV Name               /dev/dasdf1
  VG Name
  PV Size               1.15 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               vBfDc4-cZM8-mweU-CMbj-4VQQ-ktvr-CqSgTN
```

Creating a single volume group

The **vgcreate** command can be used to create a volume group named `homevg` from the two partitions. Use the **vgdisplay homevg** command to verify that the volume group was created:

```
# vgcreate homevg /dev/dasde1 /dev/dasdf1
Volume group "homevg" successfully created
# vgdisplay homevg
--- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
```

```
  Cur PV                2
  Act PV                2
  VG Size               2.29 GB
  PE Size               4.00 MB
  Total PE              586
  Alloc PE / Size       0 / 0
  Free  PE / Size       586 / 2.29 GB
  VG UUID               9XXzEI-etSq-GBBB-WZqc-OKcO-78yT-XQMJBO
```

In this example, there are 586 free physical extents.

Creating a single logical volume

The **lvcreate** command is used to create a logical volume. The **-l** flag specifies to use all free extents, 586 in this example. The **-n homelv** specifies the name of the new logical volume. The last argument, **homevg**, specifies the name of the volume group from which the logical volume will be created.

```
# lvcreate -l 586 -n homelv homevg
  Logical volume "homelv" created
```

Use the **lvdisplay** command to verify. The parameter is the full path of the logical volume, not just the logical volume name:

```
# lvdisplay /dev/homevg/homelv
--- Logical volume ---
  LV Name                /dev/homevg/homelv
  VG Name                homevg
  LV UUID                1Ofq2D-OlpV-ATef-jOxU-91iA-OOgs-CMFyDz
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                2.29 GB
  Current LE             586
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     1024
  Block device           253:3
```

Making a file system from the logical volume

Now you have a logical volume. Use the **mke2fs** command to create a file system out of it. The **-j** flag adds a journal so it will be of type ext3:

```
# mke2fs -j /dev/homevg/homelv
mke2fs 1.41.9 (22-Aug-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
150176 inodes, 600064 blocks
30003 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=616562688
19 block groups
32768 blocks per group, 32768 fragments per group
7904 inodes per group
Superblock backups stored on blocks:
```

```
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): 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.
```

The file system created from the logical volume is now ready to be mounted.

11.2.2 Updating the file system table

You could now mount the file system manually. However, if you add the mount to the file system table file, `/etc/fstab`, you can effectively test the change by using the **`mount`** command with only one argument. Make a backup copy, then add the following line to the file:

```
# cd /etc
# cp fstab fstab.orig
# vi fstab
/dev/disk/by-path/ccw-0.0.0300-part1 swap           swap     defaults
0 0
/dev/disk/by-path/ccw-0.0.0301-part1 swap           swap     defaults
0 0
/dev/disk/by-path/ccw-0.0.0100-part1 /              ext3     acl,user_xattr
1 1
/dev/system-vg/opt-lv /opt                 ext3       acl,user_xattr        1 2
/dev/system-vg/usr-lv /usr                 ext3       acl,user_xattr        1 2
/dev/system-vg/var-lv /var                 ext3       acl,user_xattr        1 2
/dev/homevg/homelv    /home                ext3       defaults              0 0
tmpfs                 /tmp                 tmpfs      defaults              0 0
proc                  /proc                proc       defaults              0 0
sysfs                 /sys                 sysfs      noauto                0 0
debugfs               /sys/kernel/debug    debugfs    noauto                0 0
devpts                /dev/pts             devpts     mode=0620,gid=5       0 0
```

Before mounting over `/home/`, check that it is empty. In this example, a non-root user was created when installing the golden image, so that user has a directory under `/home/`. If you over-mount `/home/`, this directory will be *covered*. Move it temporarily to another location. In this example, `/tmp/` is used.

```
# ls /home
mikemac
# mv /home/mikemac /tmp
# ls /home
```

Mount the `/home/` file system with one argument. Use the **`ls`** command to verify that there is no data in the logical volume except the `lost+found/` directory (created with the **`mke2fs`** command). Move the existing home directory back from `/tmp/`. Use the **`df -h`** command to verify that it is mounted:

```
# mount /home
# mv /tmp/mikemac/ /home
# ls /home
lost+found  mikemac
# df -h
Filesystem            Size  Used Avail Use% Mounted on
```

```
/dev/dasda1           372M  105M  249M  30% /
devtmpfs              122M  144K  122M   1% /dev
tmpfs                 122M     0  122M   0% /dev/shm
/dev/mapper/system--vg-opt--lv
                      372M   17M  337M   5% /opt
/dev/mapper/system--vg-usr--lv
                      2.4G  1.2G  1.1G  52% /usr
/dev/mapper/system--vg-var--lv
                      504M   62M  417M  13% /var
/dev/mapper/homevg-homelv
                      2.3G   68M  2.1G   4% /home
tmpfs                 122M  4.0K  122M   1% /tmp
```

When /home/ can be mounted with one parameter, it shows that the syntax in /etc/fstab is correct.

Note: Make sure LVM will be activated also after a reboot. Check service start with **chkconfig -A|grep boot.lvm**. If service is disabled, enable it with **chkconfig boot.lvm on**.

Since dasde and dasdf were brought online with the **dasd_configure** command, they will stay online also after a reboot. Test a reboot to verify that the new logical volume is successfully mounted over /home/.

```
# reboot

Broadcast message from root (pts/0) (Thu Sep  2 15:08:07 2010):

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

11.3 Extending an existing logical volume

This section describes the process of adding a new minidisk to an existing LVM. This is useful when your logical volume has run out of space.

First, repeat the steps described in 11.1, “Adding DASD” on page 202 to add a new minidisk. In this example, a minidisk at virtual address 104 is added of size 3338 cylinders. Do not forget to log off and log back on to LINUX02 so that the new directory entry is read.

When your system comes back, bring disk 104 online and verify that it is available:

```
# dasd_configure 0.0.0104 1
Configuring device 0.0.0104
Setting device online
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2347MB    600840
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0102   active      dasdd     94:12   ECKD  4096   1173MB    300420
0.0.0301   active      dasde     94:16   FBA   512    512MB     1048576
0.0.0103   active      dasdf     94:20   ECKD  4096   1173MB    300420
0.0.0104   active      dasdg     94:24   ECKD  4096   2347MB    600840
```

The new disk (104) is /dev/dasdg. Format the minidisk with the **dasdfmt** command and make a single partition with the **fdasd -a** command, as you did previously:

```
# dasdfmt -b 4096 -y -f /dev/dasdg
Finished formatting the device.
Rereading the partition table... ok
# fdasd -a /dev/dasdg
reading volume label ..: VOL1
reading vtoc ..........: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

Creating a physical volume

Use the **pvcreate** command to create a physical volume from the minidisk:

```
# pvcreate /dev/dasdg1
  No physical volume label read from /dev/dasdg1
  Physical volume "/dev/dasdg1" successfully created
```

Extending the volume group

Use the **vgextend** command to extend the volume group into the new physical volume. Then, use **vgdisplay** to verify that the volume group has free space.

```
# vgdisplay homevg
--- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               2.29 GB
  PE Size               4.00 MB
  Total PE              586
  Alloc PE / Size       586 / 2.29 GB
  Free  PE / Size       0 / 0
  VG UUID               9XXzEI-etSq-GBBB-WZqc-OKcO-78yT-XQMJBO
# vgextend homevg /dev/dasdg1
  Volume group "homevg" successfully extended
# vgdisplay homevg
--- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  3
  VG Access             read/write
```

```
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               4.58 GB
  PE Size               4.00 MB
  Total PE              1172
  Alloc PE / Size       586 / 2.29 GB
  Free  PE / Size       586 / 2.29 GB
  VG UUID               9XXzEI-etSq-GBBB-WZqc-OKcO-78yT-XQMJBO
```

Note that there are 586 new free physical extents (PEs).

Extend the logical volume and the file system

Now that you have free space in the volume group, you can increase the size of the existing logical volume with the **lvextend** command. The **-l** option specifies the number of extents to add. Finally, use the **ext2online** command to increase the size of the file system while it is still mounted.

You can use the **df** command to show the file system size before and after you extend it, as the following example shows:

```
# df -h /home
/dev/mapper/homevg-homelv
                      2.3G   68M  2.1G   4% /home
# lvextend -l +586 /dev/homevg/homelv
  Extending logical volume homelv to 4.58 GB
  Logical volume homelv successfully resized
# resize2fs /dev/homevg/homelv
resize2fs 1.41.9 (22-Aug-2009)
Filesystem at /dev/homevg/homelv is mounted on /home; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/homevg/homelv to 1200128 (4k) blocks.
The filesystem on /dev/homevg/homelv is now 1200128 blocks long.
```

Use the **df -h** command to show that the file system is now 2.3 GB larger:

```
# df -h /home
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/homevg-homelv
                      4.6G   69M  4.3G   2% /home
```

11.4 The X Window System

For many years UNIX-like operating systems have been using the X Window System (commonly just “X”). This system was designed to provide client/server, hardware-independent and network-enabled graphical environment. The current version is X11, which is widely used on UNIX and Linux platforms.

Confusion often arises among new X users regarding the concept of client and server, because client and server are defined from an application point of view where other protocols such as SSH, Telnet, and FTP are defined from a user point of view. In X the server runs on

the hardware with the mouse, keyboard and monitor (usually a workstation or a desktop), while the client runs on the UNIX or Linux server. Many Linux desktop users do not recognize this difference because they often run both the server and client on their desktop.

It is common practice to connect from a PC (SSH client) to remote Linux (SSH server) and then run an X application. It runs on remote Linux (X client) and displays on local PC (X server).

The X communication protocol by its nature is not secure at all. For this reason it is often used together with SSH protocol, which tunnels X11 traffic using encrypted (and thus secure) communications.

X11 itself provides the ability to display graphics on raster display, nothing more. If you want to be able to move, resize and otherwise manage windows, a *window manager* is needed. There are many window managers available; some are lightweight while some are more robust. So using a window manager is a good idea because it provides functionality that one expects from a GUI.

When you have Linux installed on your workstation, a window manager is probably not enough. Here you want a full desktop environment with menus, icons, task bars, and so on, such as Gnome and KDE.

When installing SLES11 SP1, there are three groups of RPMs available under Graphical Environment (Figure 11-2). None is really needed for a server system.

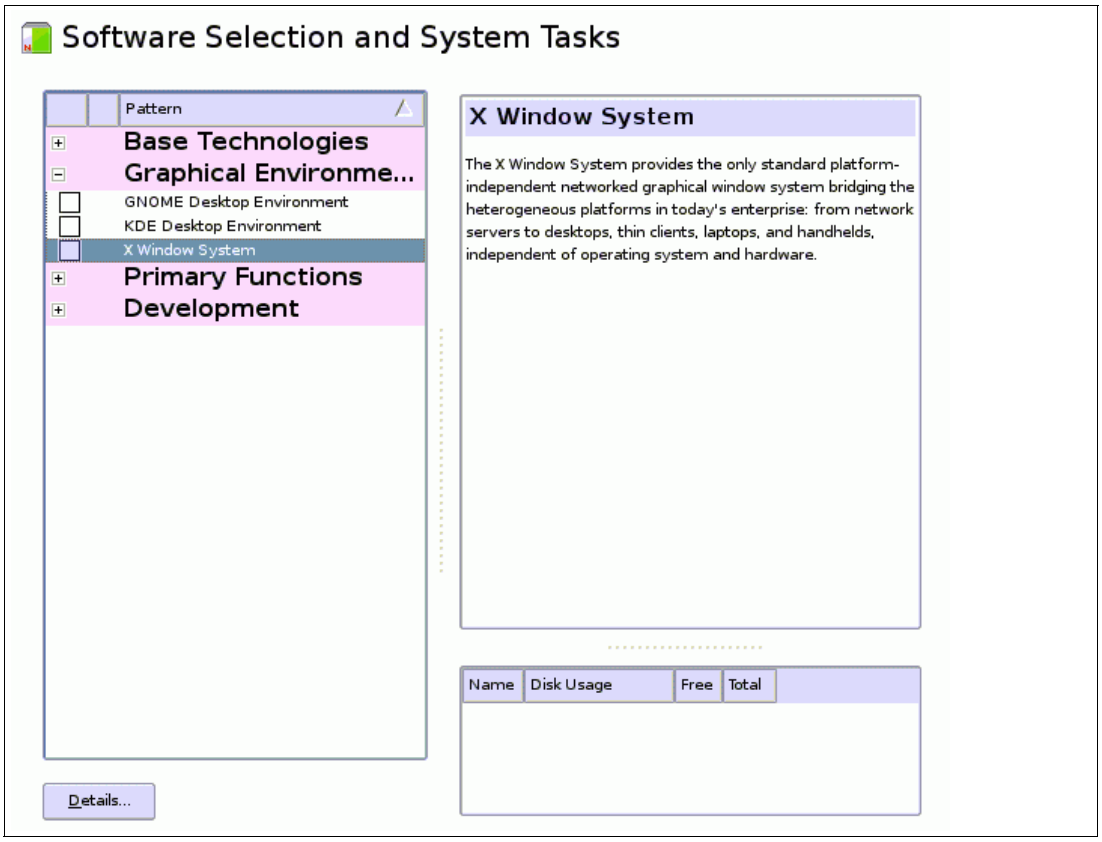


Figure 11-2 Graphical Environments in SLES11 SP1

Installing Gnome or KDE on System z is discouraged because they are resource-intensive. Installing the X Window system is also not advisable. Therefore, it is recommended to unselect them all during installation.

11.4.1 VNC server

As mentioned earlier, the X server is run where the mouse, keyboard and monitor are located —on the workstation. In a nutshell, the VNC server provides a virtual workstation with all its peripherals (virtual). The VNC server starts an embedded X server. Then any X-based application can send its output to this X server, regardless whether the application is local or remote to the X server.

To interact with the X server, one uses a VNC client on a workstation, as described in 3.2, “Setting up a VNC client” on page 23. The VNC server customization is described in 8.5.1, “Configuring the VNC server” on page 154. In our experience this is all you need if you want to run X applications from time to time.

One big advantage of VNC is that it is session oriented. If communication to the VNC server is lost, a connection is reestablished to the session as it was. Also, applications in a disconnected VNC session still continue to run.

11.4.2 X server on a workstation

If for some reason VNC is not acceptable, it is possible to use a standard X server on a workstation. Since Linux users usually know the X Window system, an X server running on Windows is described in this section.

There are many commercial and free X Window servers available for Windows. In the following examples XliveCD is used, which provides a free X server based on Cygwin. It can be run directly from a CD without requiring installation. See:

`http://xlivecd.indiana.edu/`

Any X application will send its output to an address defined with the **`-display`** parameter or, if not provided, to an address specified in the `DISPLAY` environment variable. If neither is provided, the local computer is used for output.

```
# xclock
Error: Can't open display:
```

There is no display specified for the **`xclock`** command and it will terminate.

Display is specified by setting the DISPLAY environment variable.

```
# export DISPLAY=9.145.177.158:0
# xclock
No protocol specified
Error: Can't open display: 9.145.177.158:0
```

This command failed, because the XliveCD requires an explicit command to allow remote hosts to connect to it. When the command **`xhost +`** (plus means to add authorized hosts) is run, **`xclock`** can finally display on Windows as shown in Figure 11-3 on page 213. Remember that the program itself runs on a remote Linux.

```
# xclock &
[1] 21915
```

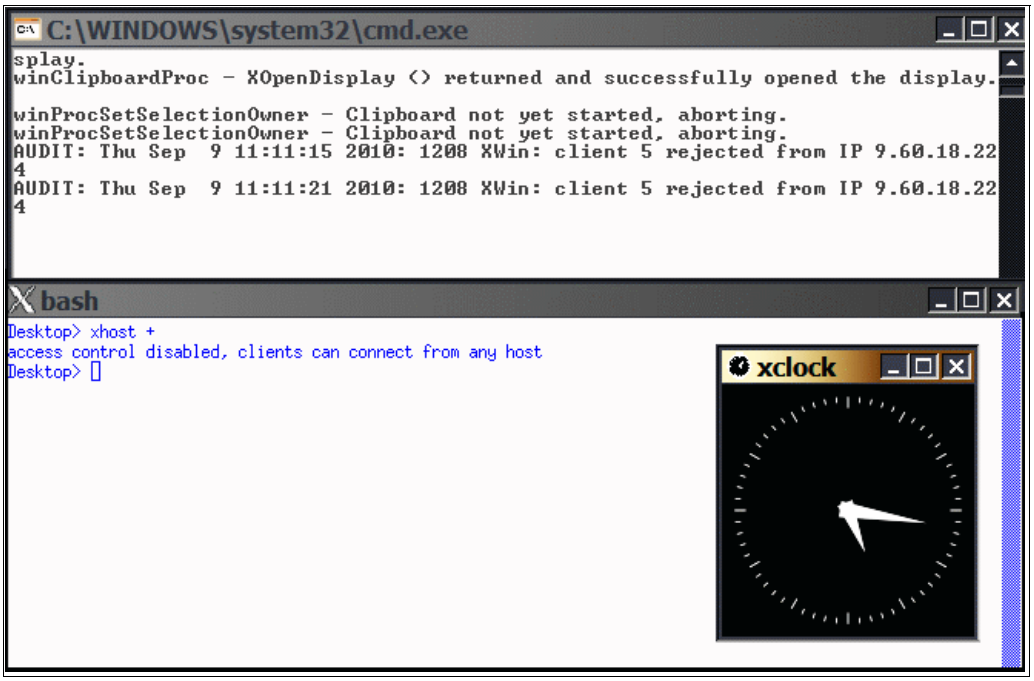


Figure 11-3 Manual setting of the DISPLAY variable

The **xhost +** command allows any host to access the X server. From a security point of view, this may not be a good idea. Even allowing only specific hosts is not enough, because X11 protocol itself is not secure. Using SSH tunneling removes this security exposure. SSH tunneling also prevents firewalls and NAT from breaking X11 communications.

It is possible to use an external SSH client that allows X11 forwarding, or an SSH client embedded in XliveCD itself. Both options are shown.

Using PuTTY

To use PuTTY for X11 forwarding, select X11 forwarding, as shown in Figure 11-4 on page 214.

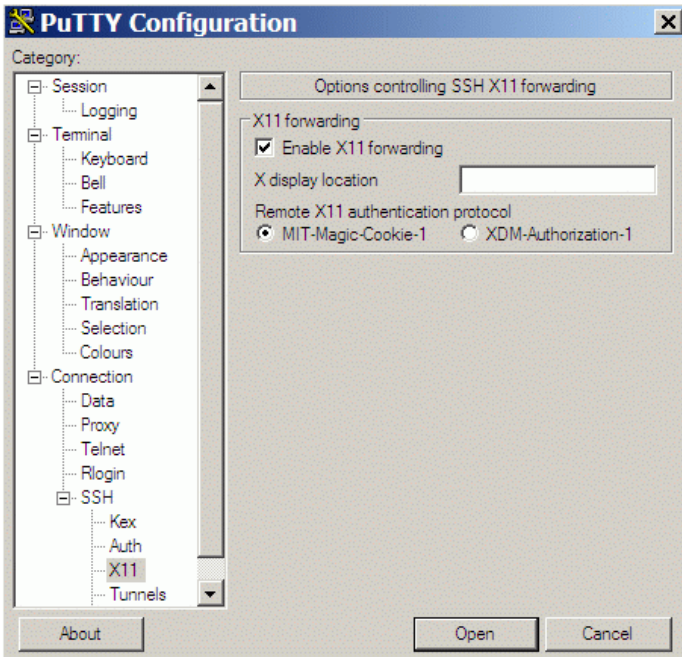


Figure 11-4 Allow X11 Forwarding in PuTTY

As you can see in Figure 11-5, the `DISPLAY` environment variable contains the special value of **`localhost:10.0`**, which tells PuTTY to forward X11 protocol over SSH to the SSH client address. In this case there is no need to use the **`xhost`** command because the connection appears to the X server as a local one.

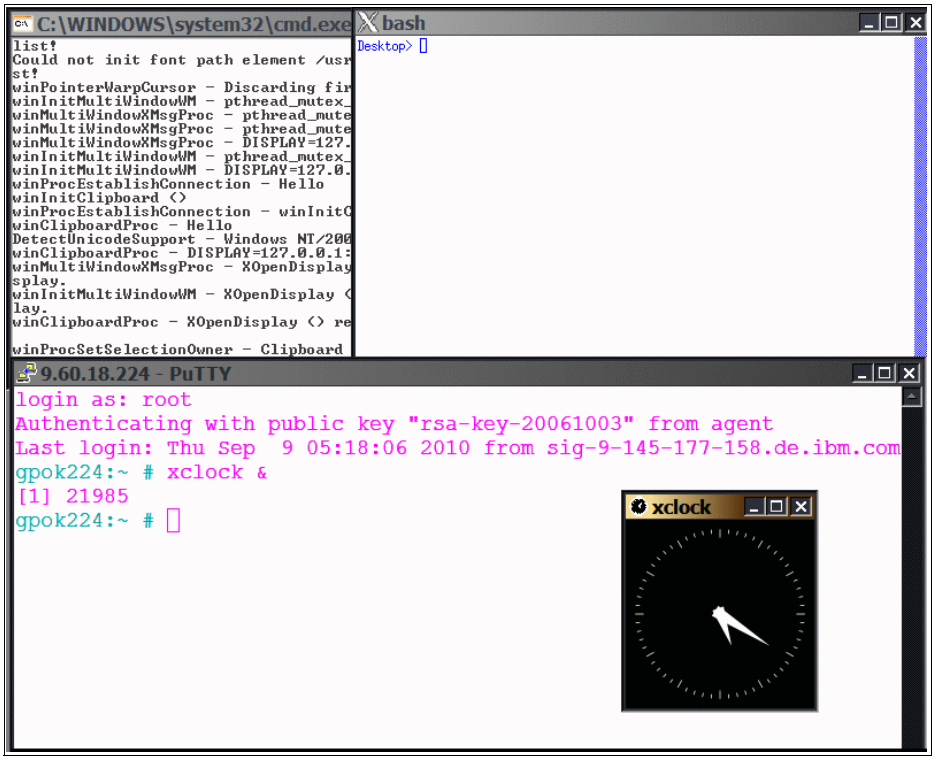


Figure 11-5 X11 forwarding with PuTTY

Using embedded SSH

It is also possible to achieve X11 forwarding with an embedded SSH client, as shown in Figure 11-6. Again, no **xhost** command is needed.

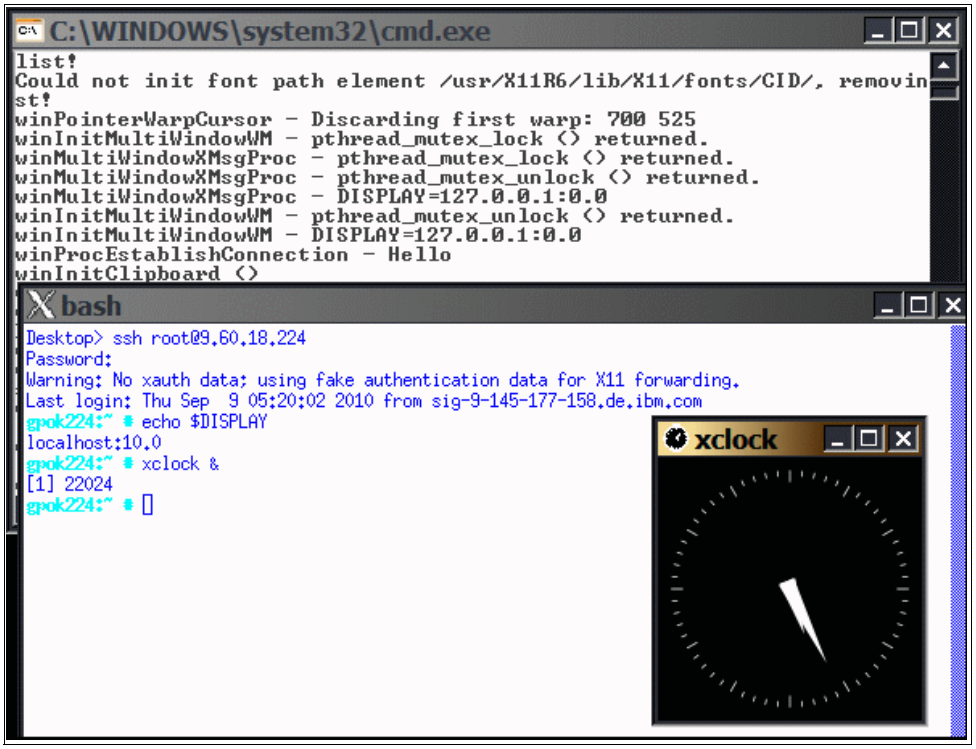


Figure 11-6 X11 forwarding with embedded SSH client

There are many ways to achieve the same results. It is up to you to choose a solution that suits your purpose best.

11.5 Setting up Memory Hotplug

Linux Memory Hotplug allows the amount of memory in a Linux system to be increased or decreased without a reboot. You must first have standby memory defined to the virtual machine in which Linux is running. You can issue the **CP DEFINE STORAGE** command to configure standby memory (storage). SLES 11 SP1 can then exploit the standby memory using the Service Call (**SERVC**) instruction.

To set up standby storage for Linux Memory Hotplug, using `LINUX01` as the virtual machine, perform the following steps.

- Modify the `LINUX01` directory entry by adding a **COMMAND** statement. This gives the virtual machine an additional 768 MB of standby memory:

```
USER LINUX01 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  COMMAND DEFINE STORAGE 256M STANDBY 768M
  OPTION APPLMON
  MDISK 100 3390 3339 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 6677 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
```

- You could run the **DISKMAP USER** command to review the minidisk allocation, but because you did not change anything to do with disks, it is probably not necessary. Run the **DIRECTXA** command to bring the change online:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 45 disk pages
```

- Shut down the Linux system running on LINUX01. This can be done a number of ways, but because you are logged onto MAINT, it can be accomplished with the **SIGNAL SHUTDOWN** command:

```
==> signal shutdown linux01
```

- In about 30 seconds, you should see notification that the system shut down cleanly and the virtual machine was logged off:

```
HCPSIG2113I User LINUX01 has reported successful termination
USER DSC   LOGOFF AS  LINUX01  USERS = 16    AFTER SIGNAL
```

- Log on to LINUX01. You should see the standby memory reported:

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES: 0003 RDR,   NO PRT,   NO PUN
00: LOGON AT 11:47:27 EDT MONDAY 09/13/10
00: STORAGE = 256M MAX = 1G INC = 1M STANDBY = 768M RESERVED = 0
00: Storage cleared - system reset.
```

- Answer **y**es to boot Linux:

```
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
00: zIPL v1.8.0-44.22.5 interactive boot menu
00:
00:  0. default (SLES11_SP1)
00:
00:  1. SLES11_SP1
00:  2. FailsafeV1
00:  3. ipl
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 3 seconds):
...
```

- Start an SSH session as root and issue the **lsmem -a** command. You should see the 768 MB of standby memory being reported as offline:

```
# lsmem -a
Address Range                       Size (MB)  State    Removable  Device
==========================================================================
0x0000000000000000-0x000000000fffffff        256  online   no         0-127
0x0000000010000000-0x000000001fffffff        256  offline  -          128-255
```

```
0x0000000020000000-0x000000002fffffff        256  offline  -          256-383
0x0000000030000000-0x000000003fffffff        256  offline  -          384-511

Memory device size  : 2 MB
Memory block size   : 256 MB
Total online memory : 256 MB
Total offline memory: 768 MB
```

- Another way to view the status of the memory is in the `/sys/` file system. Change the directory to `/sys/devices/system/memory/` and list the files:

```
# cd /sys/devices/system/memory
# ls
block_size_bytes  memory0  memory1  memory2  memory3
```

- Type the `block_size_bytes` file with the **`cat`** command:

```
# cat block_size_bytes
10000000
```

This number is the number of bytes in hexadecimal. 10000000 in hex is 256 M. So the block size is 256 MB and there are four blocks: `memory0-memory3`, which are represented as directories. Each of the memory blocks has a state, which is represented as a file.

- Show the state of each memory block with the following command:

```
# cat memory*/state
online
offline
offline
offline
```

This shows that the first 256 MB are online and the next three blocks are offline. This corresponds to the output of **`lsmem`**.

- You can turn on memory by sending the string `online` to the state file. Turn on an additional 512 MB of memory with the following commands:

```
# echo online > memory1/state
# echo online > memory2/state
```

- Show that the memory is now online with the **`lsmem -a`** command:

```
# lsmem -a
Address Range                              Size (MB)  State    Removable  Device
==========================================================================================
0x0000000000000000-0x000000000fffffff        256  online   no         0-127
0x0000000010000000-0x000000001fffffff        256  online   yes        128-255
0x0000000020000000-0x000000002fffffff        256  online   yes        256-383
0x0000000030000000-0x000000003fffffff        256  offline  -          384-511

Memory device size  : 2 MB
Memory block size   : 256 MB
Total online memory : 768 MB
Total offline memory: 256 MB
```

- You can also give the memory back by echoing `offline` to the state file:

```
# echo offline  > memory1/state
# echo offline  > memory2/state
# lsmem -a
...
Memory device size  : 2 MB
```

```
Memory block size   : 256 MB
Total online memory : 256 MB
Total offline memory: 768 MB
```

This section has shown how to configure virtual machines with standby memory and how to "hot-plug" the memory from Linux. Each of the four Linux virtual machines, LINUX01 - LINUX04, defaults to 256 MB of memory and can be moved up to 1 GB. However, LINUX02 - LINUX04 require Linux to be shut down, the **CP DEFINE STORAGE** command to be run, and Linux to be rebooted. LINUX01 can now have memory added while Linux is running. This function can increase your server farm's performance and availability.

11.6 Utilizing the cpuplugd service

The **cpuplugd** service allows Linux to enable or disable processors and memory, based on a set of rules. It can improve performance by setting the correct number of processors and amount of memory for Linux systems depending on their current load. It can also prevent the Linux scheduler from queue balancing in partial load situations.

More information on **cpuplugd** can be found in the manual *Device Drivers, Features, and Commands (kernel 2.6.34)* on the web at

http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html

11.6.1 Giving Linux virtual machines more processors

To start work with **cpuplugd**, perform the following steps:

- Log on to a 3270 session as MAINT and determine how many physical processors your LPAR has with the **QUERY PROCESSORS** command:

```
==> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
PROCESSOR 03 ALTERNATE CP
PROCESSOR 04 ALTERNATE CP
PROCESSOR 05 ALTERNATE CP
PROCESSOR 06 ALTERNATE CP
PROCESSOR 07 ALTERNATE CP
PROCESSOR 08 ALTERNATE CP
PROCESSOR 09 ALTERNATE CP
```

  In this example, there are 10.

- Start an SSH session to a Linux and determine how many processors Linux has online. Write a short bash script, **lscpus**, to save typing:

```
# cd /usr/local/sbin
# vi lscpus
#!/bin/bash
# script to list the number and status of virtual CPUs
for i in /sys/devices/system/cpu/cpu*
do
  echo $i
  cat $i/online
done
```

- Save the file and the set it to be executable:

```
# chmod +x lscpus
```

- Run the script:

```
# lscpus
/sys/devices/system/cpu/cpu0
1
```

This shows that LINUX01 has only one virtual processor. There is only one virtual processor because the LNXDFLT profile in the USER DIRECT file has only one CPU statement:

```
PROFILE LNXDFLT
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
  ...
```

For **cpuplugd** to hot plug processors, each Linux virtual machine must be given more virtual processors . To do that, perform the following steps:

- Go back to your 3270 session as MAINT. Add additional virtual processors by adding more CPU statements to the LNXDFLT profile. In this example, nine more are added so that each Linux virtual machine will have the same number of virtual processors because there are physical processors. Also, increase the maximum number of processors that can be defined to **10** on the MACHINE statement:

```
==> x user direct c
====> /profile lnxd
PROFILE LNXDFLT
  IPL CMS
  MACHINE ESA 10
  CPU 00 BASE
  CPU 01
  CPU 02
  CPU 03
  CPU 04
  CPU 05
  CPU 06
  CPU 07
  CPU 08
  CPU 09
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
  SPOOL 000C 2540 READER *
 ...
```

- Save the file and update the user directory:

```
==> directxa
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 45 disk pages
```

- Shut down LINUX01 and restart it after it reports successful termination:

```
==> signal shutdown linux01
HCPSIG2113I User LINUX01 has reported successful termination
USER DSC   LOGOFF AS  LINUX01  USERS = 15    AFTER SIGNAL
==> xautolog linux01
```

```
Command accepted
AUTO LOGON  ***       LINUX01  USERS = 16
HCPCLS6056I XAUTOLOG information for LINUX01: The IPL command is verified by
the
 IPL command processor.
```

- After Linux boots in a minute or two, start an SSH session to `LINUX01`.
- List the processors with the new **`lscpus`** script:

```
# lscpus
/sys/devices/system/cpu/cpu0
1
/sys/devices/system/cpu/cpu1
1
/sys/devices/system/cpu/cpu2
1
/sys/devices/system/cpu/cpu3
1
/sys/devices/system/cpu/cpu4
1
/sys/devices/system/cpu/cpu5
1
/sys/devices/system/cpu/cpu6
1
/sys/devices/system/cpu/cpu7
1
/sys/devices/system/cpu/cpu8
1
/sys/devices/system/cpu/cpu9
1
```

This shows that there are now ten virtual processors and that they are all active. Giving every Linux more virtual processors will not necessarily improve performance. It may well degrade it. The **`cpuplugd`** service should control the number of CPUs based on each system's workload.

Important: Once you have given each Linux more virtual CPUs, it is important that you complete the rest of this section to turn **`cpuplugd`** on.

11.6.2 Turning cpuplugd on

The **`cpuplugd`** configuration file is `/etc/sysconfig/cpuplugd`. For now, only one change is recommended to this file. Perform the following steps:

- Make a backup of the configuration file and change the `CPU_MIN` variable to a value of **1**.

Important: Some middleware products have a requirement of a minimum of two virtual processors. If the majority of your Linux servers will be running a workload that recommends two processors, leave the default for `CPU_MIN` at 2. An exception would be when only a single physical processor is available.

```
# cd /etc/sysconfig
# cp cpuplugd cpuplugd.orig
# vi cpuplugd
#
```

```
# Exemplary configuration file for the cpuhotplug daemon for
# Linux on System z
#
# The file is evaluated by cpuplugd at startup when called with -c.
# It does not contain shell environment variables.

## Type:        integer
## Default:     2
#
# The minimum number of cpus.
# This means in this example, that every time at least one cpu
# will be available
#
CPU_MIN="1"
...
```

- Save the file, then turn the **cpuplugd** service on with the **chkconfig command**:

```
# chkconfig cpuplugd on
```

- Reboot the system so the **cpuplugd** service can be started at boot time:

```
# reboot

Broadcast message from root (pts/0) (Fri Sep  3 16:15:34 2010):

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

- When the system comes back, restart the SSH session and run the **lscpus** script again:

```
# lscpus
/sys/devices/system/cpu/cpu0
1
/sys/devices/system/cpu/cpu1
0
/sys/devices/system/cpu/cpu2
0
/sys/devices/system/cpu/cpu3
0
/sys/devices/system/cpu/cpu4
0
/sys/devices/system/cpu/cpu5
0
/sys/devices/system/cpu/cpu6
0
/sys/devices/system/cpu/cpu7
0
/sys/devices/system/cpu/cpu8
0
/sys/devices/system/cpu/cpu9
0
```

This shows that there are now ten virtual processors, but only one is active. The **cpuplugd** service is doing its job. If you get a new SSH session and run the **lscpus** script quickly enough, you may see that all but one of the processors have not yet been turned off.

You have now enabled **cpuplugd** for hot plugging of virtual processors. The default rules for the plugging and unplugging of processors in the configuration file is as follows:

```
HOTPLUG = "(loadavg > onumcpus +0.75) & (idle < 10.0)"
```

```
HOTUNPLUG = "(loadavg < onumcpus -0.25) | (idle > 50)"
```

Where the variables in the statements have the following meaning:

`loadavg`	The current average CPU load
`onumcpus`	The number of CPUs that are online
`runable_proc`	The current number of processes that can be run
`idle`	The current idle percentage

11.6.3 Generating a workload to see cpuplugd work

You can now generate a workload to show that **`cpuplugd`** will turn on processors.

Important: Running the following command will generate significant processor use. Verify that there is not a mission-critical workload running on this z/VM LPAR, because this test may affect it. Also, be sure to kill the processes after seeing **`cpuplugd`** in action.

Perform the following steps:

- Put ten looping jobs in the background with the following **`for`** loop:

```
# for i in `seq 1 10`
> do
>   bash -c "cat /dev/zero > /dev/null" &
> done
[1] 2251
[2] 2252
[3] 2253
[4] 2254
[5] 2255
[6] 2256
[7] 2257
[8] 2258
[9] 2259
[10] 2260
```

- See that the jobs are running (you can also use the **`top`** command):

```
# pstree -G | grep cat
     +-sshd---sshd---bash---10*[bash---cat]
```

- Now run **`lscpus`** every so often. The following example shows that, after a minute or so, **`cpuplugd`** has started five of the nine spare processors.

```
# lscpus
/sys/devices/system/cpu/cpu0
1
/sys/devices/system/cpu/cpu1
1
/sys/devices/system/cpu/cpu2
1
/sys/devices/system/cpu/cpu3
1
/sys/devices/system/cpu/cpu4
1
/sys/devices/system/cpu/cpu5
1
/sys/devices/system/cpu/cpu6
```

```
0
/sys/devices/system/cpu/cpu7
0
/sys/devices/system/cpu/cpu8
0
/sys/devices/system/cpu/cpu9
0
```

After a few more minutes, all of the processors should be activated.

- Kill the processes with the **killall** command, then verify that the loops have stopped:

```
# killall cat
bash: line 1:  2261 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2264 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2263 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2262 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2265 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2266 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2267 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2268 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2270 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2269 Terminated              cat /dev/zero > /dev/null
[1]   Exit 143                bash -c "cat /dev/zero > /dev/null"
...
# pstree -G | grep cat
```

No output shows that the processes to create a workload have been stopped.

11.6.4 Setting memory sizes with cpuplugd

Memory sizes can also be set by the **cpuplugd** service. However, unlike processors, there is no good generic default value. The following example is in the Device Drivers book:

```
MEMPLUG = "swaprate > freemem+10 & freemem+10 < apcr"
MEMUNPLUG = "swaprate > freemem + 10000"
```

However, this is just a starting point. You should test any setting that you want to implement against a representative workload that your Linux systems will be running. Details are beyond the scope of this section.

11.7 Hardware cryptographic support for OpenSSH

This section shows how to copy a test file with OpenSSH, first without any crypto acceleration. Then crypto acceleration for OpenSSH is enabled and the same file is copied again. A much higher throughput rate should be observed. The prerequisite for using hardware cryptography is to have a firmware level of LIC 3863 installed on your System z CEC.

This section is based on the white paper *First experiences with hardware cryptographic support for OpenSSH with Linux for System z*, by Manfred Gnirss, Winfried Münch, Klaus Werner and Arthur Winterling. It is on the web at:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101690

This section only shows a single example of crypto acceleration. For a more complete analysis, see the white paper.

To test copying a file with and without cryptographic acceleration, perform the following steps:

- Create a 200 MB test file for copying in the /etc/ssl/ directory:

```
# cd /etc/ssl
# dd if=/dev/zero of=testdata.txt bs=1048576 count=200
200+0 records in
200+0 records out
209715200 bytes (210 MB) copied, 17.87 s, 11.7 MB/s
# ls -lh testdata.txt
-rw-r--r-- 1 root root 200M Sep 15 08:55 /tmp/testdata.txt
```

- Copy the file locally with the **scp** command, two times with specific encryption algorithms and once without, prefixing all with the **time** command:

```
# time scp -c 3des-cbc testdata.txt localhost:/dev/null
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is ee:5e:49:bb:a8:4e:64:83:4d:7d:96:d3:a4:ca:ac:62.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
Password:
testdata.txt                                  100%  200MB  11.1MB/s   00:18

real    0m20.327s
user    0m17.334s
sys     0m0.613s
# time scp -c aes128-cbc testdata.txt localhost:/dev/null
Password:
testdata.txt                                  100%  200MB  33.3MB/s   00:06

real    0m9.218s
user    0m5.763s
sys     0m0.583s
# time scp testdata.txt localhost:/dev/null
Password:
testdata.txt                                  100%  200MB  33.3MB/s   00:06

real    0m8.470s
user    0m5.765s
sys     0m0.591s
```

  The output shows a throughputs of about 11and 33 MB/s and a user times of about 17.3 and 5.8 seconds.

- Determine whether the necessary cryptographic-related RPMs are installed:

```
# rpm -qa | egrep "libica|ibmca"
```

  No output shows that they are not installed.

- Install the necessary RPMs with the **zypper install** command:

```
# zypper install libica openssl-ibmca openssl-ibmca-32bit
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW packages are going to be installed:
  libica-2_0_2 libica-2_0_2-32bit openssl-ibmca openssl-ibmca-32bit

4 new packages to install.
```

```
Overall download size: 200.0 KiB. After the operation, additional 849.0 KiB
will be used.
Continue? [y/n/?] (y): y
...
```

- Verify that the RPMs are now installed:

```
# rpm -qa | egrep "libica|ibmca"
libica-2_0_2-32bit-2.0.2-0.4.5
libica-2_0_2-2.0.2-0.4.5
openssl-ibmca-1.0.0-141.6.12
openssl-ibmca-32bit-1.0.0-141.6.12
```

- Verify that CP Assist for Cryptographic Function (CPACF) operations are supported:

```
# icainfo
The following CP Assist for Cryptographic Function (CPACF) operations are
supported by libica on this system:
SHA-1:    yes
SHA-256:  yes
SHA-512:  yes
DES:      yes
TDES-128: yes
TDES-192: yes
AES-128:  yes
AES-192:  yes
AES-256:  yes
PRNG:     yes
```

- Make a backup of the SSL configuration file, `/etc/ssl/openssl.cnf`:

```
# cp openssl.cnf openssl.cnf.orig
```

- Append the sample SSL configuration file under `/usr/share/doc/` to the actual SSL configuration file, `/etc/openssl.cnf`:

```
# cat /usr/share/doc/packages/openssl-ibmca/openssl.cnf.sample >> openssl.cnf
```

- Edit the appended file and search for the line with the `openssl_conf` variable. Move that line from the bottom to the top and save the file, as shown in the following example:

```
# vi openssl.cnf
/openssl_conf
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd
openssl_conf = openssl_def
...
```

- Rerun the same **scp** commands:

```
# time scp -c 3des-cbc testdata.txt localhost:/dev/null
Password:
testdata.txt                                  100%  200MB  66.7MB/s   00:03

real    0m5.890s
```

```
user    0m1.542s
sys     0m0.558s
# time scp -c aes128-cbc testdata.txt localhost:/dev/null
Password:
testdata.txt                                  100%  200MB  66.7MB/s   00:03

real    0m6.287s
user    0m0.993s
sys     0m0.541s
# time scp testdata.txt localhost:/dev/null
Password:
testdata.txt                                  100%  200MB  66.7MB/s   00:03

real    0m4.839s
user    0m0.996s
sys     0m0.548s
```

- Delete the test file:

```
# rm testdata.txt
```

You should see a much improved througput. The `user` field of the **`time`** command may be the best gauge of the throughput. Based on that field, this example shows throughput increases of approximately 91% for the `3des-cbc` cipher and 83% for the `aes128-cbc` cipher. The throughput increase when no cipher is specified on the **`scp`** command seems to show that the default is to use `aes128-cbc` encryption.

11.8 Centralizing home directories for LDAP users

In previous versions of this book there was a section on how to create a travelling `/home/` directory using LDAP, NFS, and automount. In the interest of time, this section has been removed. The recipe for SLES 10 SP2 should be useful on SLES 11 SP1, though there is no promise it will work identically. See 11.3 in the IBM Redbooks publication *z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2*, SG24-7493, on the web at:

`http://www.redbooks.ibm.com/abstracts/sg247493.html`

However, the following section has been added to this book. In December of 2009, the topic of how to set up a common home directory came up on the linux-390 list server. The following post by Patrick Spinler is copied, with permission, as it may be helpful to you:

11.8.1 Recommendations for centralizing home directories

“NFSv3 is not known for its security. Consider the use of the NFS option `root_squash`, along with limiting the list of hosts who can connect to your home share. Only export home directories to hosts which you control, remember that anyone who has root on their box (e.g. a developer workstation) can impersonate any user to NFS. Here's the relevant `/etc/exports` line we use:

```
/export/unixdata/homedirs                \
  @hgrp_autohome_admin(rw,no_root_squash,insecure,sync) \
  @hgrp_autohome_hosts(rw,root_squash,insecure,sync)
```

I look forward to going to NFSv4 with kerberos authentication, but we're not there yet.

Regarding automount maps in LDAP, this works very well for us with one exception. The problem is that there's a significant number of automount map schemas out there, and different OS's (and different revisions of OS's) use different ones. As we are a fairly heterogeneous environment, I found it near impossible to keep a master map in LDAP. Right now we're just keeping a `/etc/auto.master` or `/etc/auto_master` on each host.

In order to make the individual map entries work heterogeneously, I had to add several object classes and a few redundant attributes to each entry. Here's what my home directory automount map entry looks like:

```
# ap00375, auto_home, unix.example.com
dn: automountKey=ap00375,automountMapName=auto_home,dc=unix,dc=example,dc=com
automountInformation: linux01.example.com:/vol/vol2/unixhomes-5gb/75/ap00375
cn: ap00375
automountKey: ap00375
objectClass: automount
objectClass: nisNetId
objectClass: top
```

Regarding heterogeneous clients, we found AIX in particular to be the hardest of our clients to configure, and Linux the easiest. Insure on AIX that you have the latest available LDAP client package from IBM. Also be aware that AIX wants to use it's extended LDAP schema rather than RFC2307, and wants full write access to the LDAP servers from every AIX client. Despite that, it will work with RFC2307 and read only access. Solaris, like Linux, has an option to not use an LDAP proxy account at all via anonymous binding, but I never got Solaris anonymous binding to work.

I recommend making LDAP use TLS or SSL on the wire, in order to keep clear-text passwords from flying about. Both AIX and Solaris require the server public SSL certificates to be loaded on every client to do LDAP over TLS or SSL. Linux can be configured to ignore authenticating the LDAP servers' certificates and proceed with TLS/SSL anyway - this is convenient, but does open the possibility of man in the middle attacks. In our environment this isn't a big deal, but it might be in yours.

We've found POSIX group membership management to be one of our more challenging issues overall. Some older systems (e.g. solaris <= 8 or 9) enforce the old POSIX limit of no more than 16 secondary groups. Further, the primary group concept is annoying - conceptually, in any organization with modest member mobility, which primary group do they get? If one assumes that the primary group is meaningful, e.g. reflective of someone's function, role, or job, what about people who do two or more things (E.g. student *and* employee) or people who transfer, but will have a transitional period?

Our not so great compromise was to first use NIS-style netgroups via LDAP for anything we can. In particular, we use a mutation of netgroups to control individual's authorization to log in via the use of service search descriptors, and also for sudo privileges. Second in our environment all meaningful POSIX groups are secondary groups. For primary groups we adopt the Linux convention of creating a separate POSIX group for each individual: e.g. userA gets a group userA as her primary group. This has the problem of a huge proliferation of groups, though, and several LDAP clients, in particular AIX, have issues with that."

12

Monitoring and tuning z/VM and Linux

This chapter briefly describes how to monitor z/VM and Linux. For another source 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/epubs/pdf/hcsx0b20.pdf`

There are a number of z/VM monitoring tools such as CA’s VM:Monitor, the IBM z/VM Performance Toolkit, the IBM Tivoli OMEGAMON® XE for z/VM and Linux. and products from Velocity Software. The IBM z/VM Performance Toolkit is briefly described in this section.

There are also two sections on tuning z/VM and Linux using Cooperative Memory Management (CMM) and the processor plug daemon, **`cpuplugd`**.

This chapter contains the following sections:

12.1 Using INDICATE and other commands

z/VM has many commands to monitor the state of the system. **CP INDICATE** is the most commonly used, and there are other commands that are addressed. For more information, see the z/VM Performance Resources Web page at

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

12.1.1 Using the INDICATE command

z/VM has some basic commands such as **INDICATE**. There are many **INDICATE** parameters that can be included as command line options. Use the command **HELP INDICATE** for a basic understanding and then press F11 for help on each parameter.

INIDICATE LOAD

If no parameter is specified, **INDICATE LOAD** is the default option. There are two flavors of this, depending on whether the issuing user ID has privilege class G or class E. Class G users can use **INDICATE** to display recent contention for system resources, environment characteristics, and measurements of resources used by their virtual machine.

The output from a user ID with class E privilege (for example, MAINT, OPERATOR) is shown here. The lines are numbered for clarity of the description that follows:

```
==> 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 on which 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 on-line 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 users ready to run virtual processors to keep the physical processors busy. One of the processors will be a Master, all of the others Alternate, and some imbalance may result from performing these functions.

  Line 2 describes paging to expanded storage. Most z/VM systems on z9 class machines can sustain several 1000s of this type of paging operations a second without any problems. z10 class machines will perform even better. The MIGRATE rate is the number of pages per second being moved from expanded storage out to paging space on DASD. A

healthy system will have a MIGRATE rate significantly lower than the XSTORE rate, probably being measured in 100s rather than 1000s. The higher values seen tend to build up over time, and are sustained over periods of intense system activity. However, there are times when the MIGRATE value may spike for brief periods of time.

- 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. For a system that has an appreciably high I/O rate, composed of reads plus writes, and a high proportion of reads, and a good hit ratio for those reads (tending to 90% or greater), the real, physical I/O avoidance can be very high. This author has seen the avoidance as high as 50% in some cases. Conversely, however, a high HIT RATIO with a low value for the READS rate should not be taken as good (100% hit ratio, when doing only 1 I/O per second is effectively meaningless).
- Line 4 describes more storage (memory) management. The PAGING rate is important. Higher values will often impact performance. This can be at least partially offset by increasing the number of page volumes, but a more thorough examination of this problem is advisable whenever it arises.The STEAL percentage is often misleading. This is basically the percentage of pages taken from guests that z/VM believes are non-dormant. 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 still considered to be stolen. So there are scenarios where a system only has a user set comprising active guests, in which case all pages taken would be considered stolen. Bearing this in mind, if a high STEAL value is observed, the paging rate needs to be checked. If the paging rate is relatively low, then the STEAL value is not important.
- On lines 5 through 8 you also see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into 3 different classes (1 through 3) and a special additional class labelled 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 overcommitting resources. Such a system would require further investigation, possibly leading to some tuning work, or even hardware addition in extreme cases. Ignore the values in parenthesis.

INDICATE QUEUES EXP

Another useful command to understand the state of the system is the **INDICATE QUEUES EXP**. Following is an example:

```
==> ind q exp
DATAMGT1      Q3 AP  00000537/00000537 .... -2.025 A02
BITNER        Q1 R00 00000785/00000796 .I.. -1.782 A00
EDLLNX4       Q3 PS  00007635/00007635 .... -1.121 A00
TCPIP         Q0 R01 00004016/00003336 .I.. -.9324 A01
APCTEST1      Q2 IO  00003556/00003512 .I.. -.7847 A01
EDLWRK20      Q3 AP  00001495/00001462 .... -.6996 A01
EDL           Q3 IO  00000918/00000902 .... -.2409 A01
EDLWRK11      Q3 AP  00002323/00002299 .... -.0183 A00
EDLWRK18      Q3 IO  00001052/00000388 .... -.0047 A00
EDLWRK4       Q3 AP  00004792/00002295 ....  .0055 A01
EDLWRK8       Q3 AP  00004804/00004797 ....  .0089 A02
EDLWRK16      Q3 AP  00002378/00002378 ....  .0170 A02
EDLWRK2       Q3 AP  00005544/00002956 ....  .0360 A00
EDLWRK12      Q3 AP  00004963/00002348 ....  .0677 A01
EDLWRK6       Q3 IO  00000750/00000302 ....  .0969 A02
EDLWRK3       Q3 AP  00005098/00005096 ....  .0999 A02
```

```
EDLWRK17      Q3 AP  00004786/00004766 ....  .1061 A01
EDLWRK9       Q3 AP  00002372/00002334 ....  .1107 A02
EDLWRK5       Q3 IO  00002376/00002376 ....  .1205 A01
EDLWRK14      Q3 AP  00002426/00002323 ....  .1238 A02
EDLLIB19      Q3 IO  00001226/00001100 ....  .1309 A02
EDLWRK19      Q3 AP  00002322/00002298 ....  .1705 A00
EDLWRK15      Q3 AP  00002839/00002781 ....  .2205 A02
EDLWRK1       Q3 AP  00002969/00002935 ....  .2491 A02
```

This is another class E command and displays the virtual processors associated with a given user ID (a single virtual machine may have multiple virtual processors), what queue (dispatch list, eligible list, or limit list) they are in, and what state they are in. This is a snapshot in time. Again you want to check this output to make sure there are no virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Q_x (x=1,2,3). The eligible list would be marked as E_x.

The third column in the example also gives the state of the virtual processor. This can be helpful to get an idea of how the virtual processors might be constrained. Virtual processors that are actually running in the snapshot period are marked with `RNN` where `NN` is the processor number they are on. An `R` without a number means the virtual processor is ready to run but there is no 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`**. One does not have to be concerned about the other columns unless detailed analysis is required or when IBM support requests it. Also, always remember that that is just a snapshot in time, so repeating this command often over time can give a more accurate picture of your z/VM system. A single snapshot cannot be regarded as indicative.

12.1.2 Using other basic commands

Some other useful basic commands are briefly mentioned. 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 is hard to find help for exactly the command you are looking for. Some useful **`HELP`** commands are as follow

```
==> help             // for basic help
==> help menus       // for menu of all z/VM help menus
==> help cp menu     // for a menu of all CP commands
==> help cpquery     // for a menu of all CP QUERY command
==> help cpset       // for a menu of all CP SET commands
```

Determining who is logged on

To see who is logged on to the system, use the **`QUERY NAMES`** command. For example:

```
==> q n
FTPSERVE - DSC , LINUX04  - DSC , LINUX03  - DSC , LINUX02  - DSC
LINUX01  - DSC , S11S1CLN - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC
VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
MAINT    -L0003
VSM     - TCPIP
```

Determining storage or memory

To see how much central and expanded storage (memory) is installed and allocated to a system, use the **QUERY STORAGE** and **QUERY XSTOR** commands. For example:

```
==> q stor
STORAGE = 16G CONFIGURED = 16G INC = 256M STANDBY = 0  RESERVED = 0
==> q xstor
XSTORE= 2048M online= 2048M
XSTORE= 2048M userid= SYSTEM usage= 0% retained= 0M pending= 0M
XSTORE MDC min=0M, max=0M, usage=0%
XSTORE= 2048M userid=  (none)  max. attach= 2048M
```

Determining processors

To see how many processors (CPs, IFLs, CPUs) you have allocated at the system level, use the **QUERY PROCESSORS** command. For example:

```
==> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
PROCESSOR 03 ALTERNATE CP
PROCESSOR 04 ALTERNATE CP
PROCESSOR 05 ALTERNATE CP
PROCESSOR 06 ALTERNATE CP
PROCESSOR 07 ALTERNATE CP
PROCESSOR 08 ALTERNATE CP
PROCESSOR 09 ALTERNATE CP
```

Determining the software level

To determine what level of CP your system is at, use the **QUERY CPLEVEL** command. For example:

```
==> q cplevel
z/VM Version 6 Release 1.0, service level 0901 (64-bit)
Generated at 09/11/09 16:51:48 EDT
IPL at 08/31/10 08:44:19 EDT
```

Determining system cylinder allocation

The **QUERY ALLOC MAP** command shows you the system allocation of spool, paging, and directory space. For example:

```
==> q alloc map
EXTENT     EXTENT                          % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
610RES 6280          1         20     20      1      1   5% DRCT ACTIVE
UV6281 6281          1       3338 600840  75482  75533  12% SPOOL
UV6282 6282          1       3338 600840      0      0   0% PAGE
UP6285 6285          0       3338 601020      0      0   0% PAGE
UP6286 6286          0       3338 601020     16     59   1% PAGE
UP6287 6287          0       3338 601020      0      0   0% PAGE
```

Determining DASD, OSA, and virtual resources

The **QUERY DASD** and **QUERY DASD FREE** commands show you what DASD is assigned to the system and what DASD is free to be assigned. Similarly, the **QUERY OSA** and **QUERY OSA FREE** commands report on the OSA resources. Finally, the **QUERY VIRTUAL ALL** command can be

useful. The following list gives the short form of these commands without any of the associated output shown:

```
==> 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 Guide*, SC24-6156, *z/VM Performance Toolkit Reference*, SC24-6157, on the web starting at the z/VM 5.4 bookshelf:

  http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61

  Search for `Toolkit` on that page.
- *The Program Directory for Performance Toolkit for VM*, GI10-0785, at:

  http://www.vm.ibm.com/progdir/6vmptk10.pdf
- *Linux on IBM zSeries and S/390: Performance Toolkit for VM*, SG24-6059, 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:

- "Configuring the z/VM Performance Toolkit"
- "Using the z/VM Performance Toolkit"

12.2.1 Configuring the z/VM Performance Toolkit

The Performance Toolkit is installed with z/VM. Configuration is described in the Program Directory. Following is a summary of how to turn it on. Again, you should configure the product only if you have ordered it.

- Query which priced products are enabled with the **`QUERY PRODUCT`** command:

```
==> q product
Product  State    Description
6VMDIR10 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FL 610
6VMPTK10 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
6VMRAC10 Disabled 00/00/00.00:00:00.$BASEDDR RACF for VM
6VMRSC10 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking Version 6 Release
1
 Modification 0
```

- To enable the z/VM Performance Toolkit, log on to `MAINT` and enter the following command:

```
==> service perftk enable
VMFSRV2760I SERVICE processing started
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
```

```
VMFSRV2760I SERVICE processing completed successfully
```

You should see a few screens of messages scroll by and finally the success messages shown above. This will enable the Performance Toolkit for the current z/VM session.

- At IPL time the `SYSTEM CONFIG` file is modified by having a line appended to the end. Verify that this has been added by the **`SERVICE`** command with the following commands:

```
==> link * cf1 cf1 rr
==> acc cf1 f
DMSACP723I F (CF1) R/O
==> x system config f
====> bot
====> -2
====> pre off
...
PRODUCT PRODID 6VMPTK10 STATE ENABLED DESCRIPTION '12/17/09.15:35:41.MAINT
PE

RFKIT Minidisk Install and Service'
```

The Performance Toolkit is now enabled. You can also verify this with the **`QUERY PRODUCT`** command again.

12.2.2 Configuring web browser support

Once the product is enabled, the TCPIP profile must be modified to enable web access to the Performance Toolkit. The following example sets the port to 80, the default for a web browser:

- Log on to `TCPMAINT`. Edit the TCPIP configuration file. In this example it is the `POKSND61 TCPIP D` file (assuming you modified this file name earlier; the default name is `PROFILE TCPIP`) and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved.

```
==> x poksnd61 tcpip d
====> /port
```

- Add the following line under the PORT entries:

```
...
PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
; 25   TCP SMTP                ; SMTP Server
; 53   TCP NAMESRV             ; Domain Name Server
; 53   UDP NAMESRV             ; Domain Name Server
; 67   UDP DHCPD               ; DHCP Server
; 69   UDP TFTPD               ; TFTPD (Trivial FTP) Server
; 69   UDP TFTPD               ; TFTPD (Trivial FTP) Server
  80   TCP PERFSVM             ; Performance Toolkit
; 111  TCP PORTMAP             ; Portmap Server
...
```

Save your changes. The `TCPIP` user ID needs to be recycled in order for our changes to take effect. You can **`FORCE`** and **`XAUTOLOG TCPIP`** from a console. Alternatively, if you are in a position to re-IPL the system, you can do that (`shutdown reipl iplparms cons=sysc`).

- When the system comes back, log on to TCPMAINT and check whether everything was successful with the **NETSTAT CLIENTS** command. You want to see that the service PERFSVM is a client (listening). This should be shown after a few windows of output:

```
==> netstat clients
...
Client: PERFSVM                Authorization: {none}
Notes Handled: none
Last Touched:   0:01:22
Vmcf error count: 0
```

The entry for PERFSVM should be at the end of the output.

12.2.3 Configuring PERFSVM

The PERFSVM user ID is the Performance Toolkit service machine.

- Log on 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 panel:

```
FCX001                Performance Toolkit for VM               Autoscroll 12
  FCXBAS500I Performance Toolkit for VM FL610
  Monitor event started -- recording is activated
  Monitor sample started -- recording is activated
  FCXPMN446E Incomplete monitor data: SAMPLE CONFIG size too small
```

- Press F12 twice to get to a CMS prompt.
- Copy the PROFILE XEDIT from the MAINT 191 disk so that editor sessions will have a common interface among user IDs.

  a. Use the **VMLINK** command to both link the disk read-only and access it as the highest available file mode. The default read password is read. However, if you changed your passwords as described in 4.9.1, “Changing passwords in USER DIRECT” on page 64, it will be lnx4vm (or whatever you set it to).

  ```
  ==> vmlink maint 191
  ENTER READ PASSWORD:
  lnx4vm
  DMSVML2060I MAINT 191 linked as 0120 file mode Z
  ```

  b. Copy the PROFILE XEDIT to the A disk:

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

- Copy the default configuration files, which are on PERFSVM's D disk, to your A disk:

```
==> copy * * d = = a
```

- The main configuration file is FCONX $PROFILE. Edit this file and search for the string VMCF.

```
==> x fconx $profile
====> /vmcf
```

  This should take you to line 175 where the next 4 lines are comments starting with an *. Perform the following changes:

  – Uncomment the second and fourth lines by changing *C to FC.
  – Change port 81 to 80 on the fourth line. This will enable you to use a browser interface without having to specify port 81 on the URL (with a :81 suffix).

  The modified lines should be as follows. Save your changes with the **FILE** subcommand:

```
*    Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
```

```
*    Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 80
*    Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
====> file
```

- Create a remote data retrieval authorization file with your z/VM system identifier (replace ***POKSND61*** with your system identifier):

```
==> x fconrmt authoriz
====> a 2
POKSND61 PERFSVM  S&FSERV
POKSND61 MAINT    DATA CMD EXCPMSG
```

- Create a system identification file that links your z/VM system and PERFSVM to the special resource name FCXRES00 (replace ***POKSND61*** with your system identifier):

```
==> x fconrmt systems
====> a
POKSND61 PERFSVM  ESA   N   FCXRES00
```

- Edit the PROFILE EXEC file, search for the word once and uncomment the five MONITOR SAMPLE and the two MONITOR EVENT statements:

```
==> x profile exec a
====> /once
```

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 NETWORK'
'CP MONITOR SAMPLE ENABLE APPLDATA ALL'
'CP MONITOR EVENT  ENABLE STORAGE'
'CP MONITOR EVENT  ENABLE I/O ALL'

'PERFKIT'                   /* Invoke the PERFKIT module   @FC012BD*/
```

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

- Set the PERFSVM virtual machine to be started at z/VM IPL time. Edit the PROFILE EXEC on AUTOLOG1 so that PERFSVM is automatically started at IPL time. First, log on to AUTOLOG1.
- Before pressing Enter at the VM READ prompt, type acc **(noprof** so that the PROFILE EXEC is not run.

```
LOGON AUTOLOG1
z/VM Version 6 Release 1.0, Service Level 1002 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 14:51:02 EDT THURSDAY 10/07/10
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V6.1.0    2010-09-23 11:31
==> acc (noprof
```

- Add a line so the virtual machine PERFSVM is started at z/VM IPL time:

```
==> x profile exec
/***************************/
/* Autolog1 Profile Exec   */
/***************************/
'cp xautolog tcpip'          /* start up TCPIP */
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'cp xautolog perfsvm'        /* start Performance Toolkit */
'cp set pf12 ret'            /* set the retrieve key */
...
```

- Save the file and log off from AUTOLOG1.

12.2.4 Increasing the size of the MONDCSS DCSS

The DCSS named MONDCSS shipped with z/VM 6.1 is often not large enough, especially when your LPAR has access to many devices. To increase the size of the DCSS, first determine where the current MONDCSS is located by entering the following command:

```
==> q nss name mondcss map
FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0011 MONDCSS  CPDCSS     N/A    09000  09FFF   SC  R  00001   N/A       N/A
```

In this example, the DCSS starts at x9000 and ends at x9FFF. This is x1000 or 4096 pages. Since a page is 4096 bytes or 4 K, the size of this DCSS is 16 MB (4 KB * 4 KB) .

Before starting the Performance Toolkit, you may want to increase the size of the DCSS named MONDCSS. The following example quadruples the size of MONDCSS to 64 MB:

- Delete the old MONDCSS with the **PURGE NSS** command:

```
==> purge nss name mondcss
NO FILES PURGED
0001 FILE  PENDING PURGE
```

- Verify that the device addresses 4000-7FFF are free with the **QUERY NSS MAP** command:

```
==> q nss map
FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0033 CMS      NSS      0000256K 00000  0000D   EW  A  00007   00-15     NO
                                00020  00023   EW
                                00F00  013FF   SR
0032 NLSKANJI DCSS       N/A    02000  020FF   SR  A  00000   N/A       N/A
0031 NLSUCENG DCSS       N/A    02000  020FF   SR  A  00000   N/A       N/A
0030 NLSAMENG DCSS       N/A    02000  020FF   SR  A  00004   N/A       N/A
0029 HELPSEG  DCSS       N/A    00C00  00CFF   SR  A  00000   N/A       N/A
0016 SCEEX    DCSS       N/A    02100  028FF   SR  A  00000   N/A       N/A
0023 ZCMS     NSS      0000256K 00000  0000D   EW  A  00000   00-15     NO
                                00020  00023   EW
                                00F00  013FF   SR
0002 GCS      NSS      0000256K 00000  0000C   EW  R  00000   OMITTED   YES
                                00400  0044E   SR
                                0044F  0044F   SW
                                00450  005FF   SN
                                01000  0101A   SR
                                0101B  011FF   SN
0018 PERFOUT  DCSS       N/A    08A00  08FFF   SN  A  00000   N/A       N/A
0017 SCEE     DCSS       N/A    00900  009FF   SR  A  00000   N/A       N/A
0014 CMSDOS   DCSS-M     N/A    00B00  00B0C   SR  A  00000   N/A       N/A
0013 CMSBAM   DCSS-M     N/A    00B0D  00B37   SR  A  00000   N/A       N/A
0012 DOSBAM   DCSS-S     N/A    00B00  00B37   --  A  00000   N/A       N/A
0010 GUICSLIB DCSS       N/A    01F00  01FFF   SR  A  00000   N/A       N/A
0009 CMSFILES DCSS       N/A    01900  01BFF   SR  A  00003   N/A       N/A
0008 SVM      DCSS       N/A    01900  019FF   SR  A  00000   N/A       N/A
0007 CMSPIPES DCSS       N/A    01800  018FF   SR  A  00011   N/A       N/A
0006 CMSVMLIB DCSS       N/A    01700  017FF   SR  A  00011   N/A       N/A
0005 INSTSEG  DCSS       N/A    01400  016FF   SR  A  00011   N/A       N/A
0003 DOSINST  DCSS       N/A    00900  0090F   SR  A  00000   N/A       N/A
```

- Redefine the DCSS to a larger size with the following **DEFSEG** and **SAVESEG** commands:

```
==> defseg mondcss 4000-7fff sc rstd
HCPNSD440I Saved segment MONDCSS was successfully defined in fileid 0034.
==> saveseg mondcss
HCPNSS440I Saved segment MONDCSS was successfully saved in fileid 0034.
```

- Verify that the new DCSS was created:

```
==> q nss name mondcss map
FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0034 MONDCSS  CPDCSS     N/A    04000  07FFF   SC  R  00000   N/A       N/A
```

You should now be ready to run the Performance Toolkit.

12.2.5 Starting the z/VM Performance Toolkit

To start the Performance Toolkit, perform the following steps:

- Log on to the PERFSVM user ID.
- Press Enter and the performance toolkit should start through the PROFILE EXEC:

```
FCX001                Performance Toolkit for VM              Autoscroll 12
 FCXBAS500I Performance Toolkit for VM FL610
 FCXAPP530I Connected to *IDENT for resource FCXRES00
 FCXAPF530I Connected to *IDENT for resource FCXSYSTM
```

```
FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
FCXAPP527I User PERFSVM connected on path 0006
FCXAPC535I Connected to resource FCXRES00 on path 0005, for S&F-Coll
FCXTCP575I WebServer host IP address is 9.60.18.249:00080
FCXTCP590I WebServer interface activated
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

Disconnect from `PERFSVM` now.

```
Command ===> disc
```

The Performance Toolkit should now be configured and running.

12.2.6 Using the z/VM Performance Toolkit

The Performance Toolkit can be used with a web browser or 3270 interface.

Using a web browser interface

To use the web-enabled Performance Toolkit, perform the following steps:

- Point a browser to your z/VM system. For example:

  `http://9.60.18.249`

- You should see a splash window, then the Web Server Logon panel, as shown in Figure 12-1:

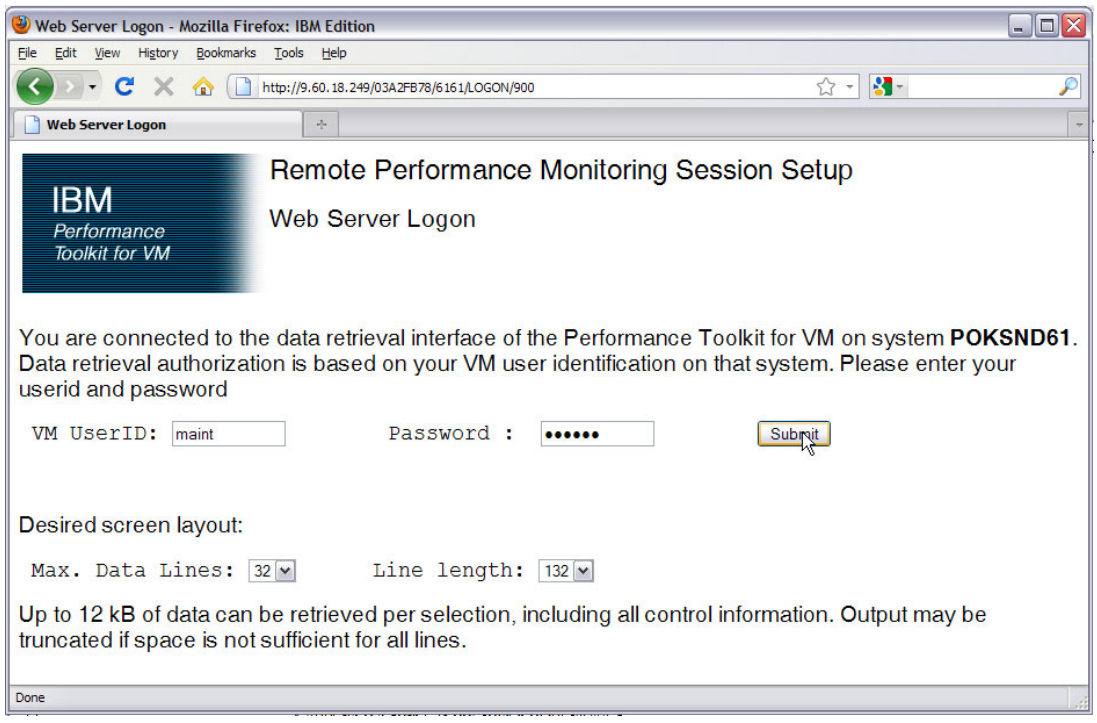


Figure 12-1 Performance Toolkit logon panel

- Enter any valid user ID and password and click **Submit**. In this example, `MAINT` is used.
- The Central Monitoring System Load Overview appears with your system identifier (Node-ID) on the left side.

- Click your system identifier and the Initial Performance Data Selection Menu window appears, as shown in Figure 12-2.
- From this window, you can drill down into many different types of reports.

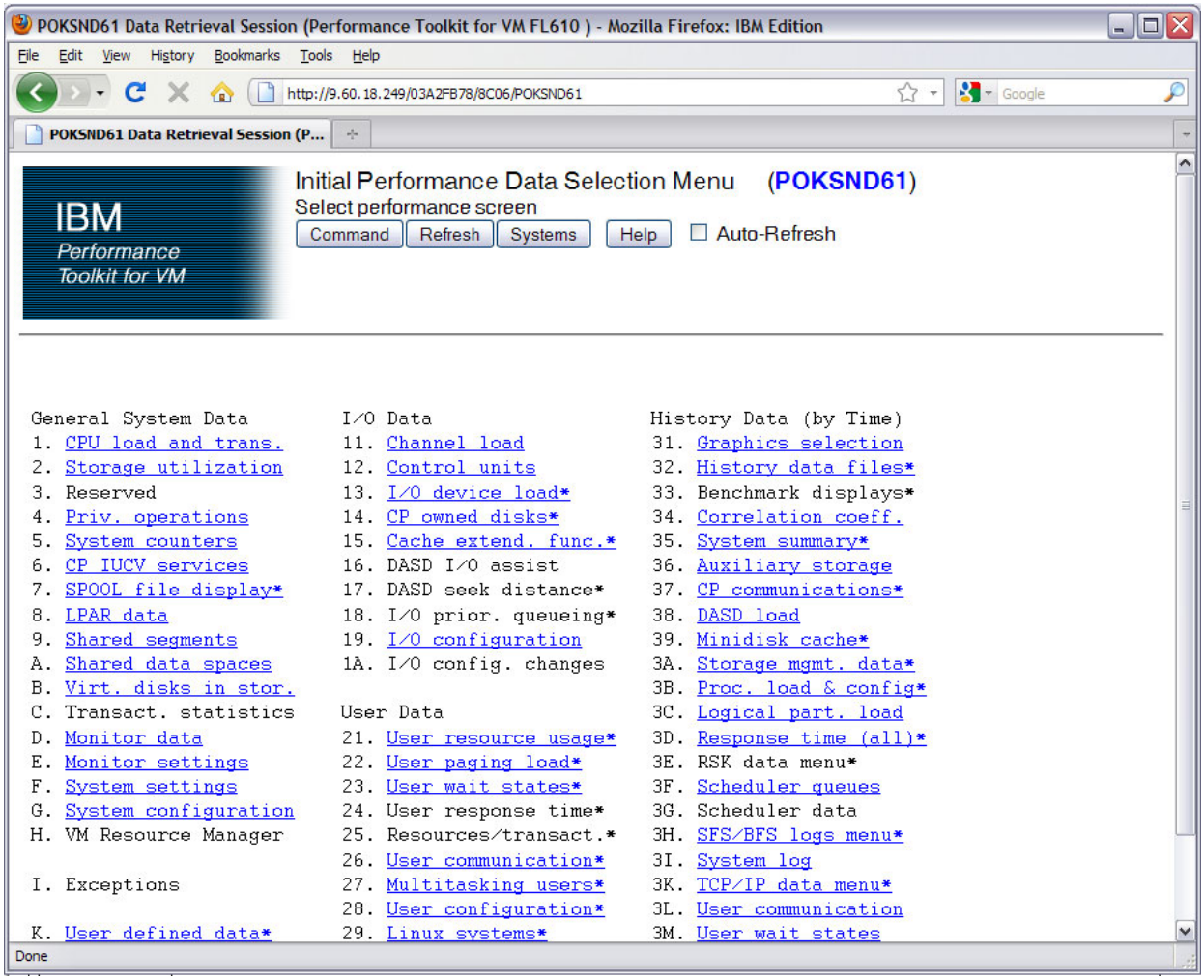


Figure 12-2 Browser interface to the Performance Toolkit

Using a 3270 interface

You can also use a 3270 interface as well as a browser interface. To do so, perform the following steps:

- Log on to PERFSVM.
- If you had disconnected, pressing Enter should get you back to the Performance Toolkit command line. If the virtual machine was logged off, the PROFILE EXEC should run and get you to the command line; see Figure 12-3 on page 242. Enter the command **MONITOR**:

  ```
  Command ==> monitor
  ```

```
FCX124          Performance Screen Selection  (FL610       )    Perf. Monito

General System Data        I/O Data                  History Data (by Time)
1. CPU load and trans.     11. Channel load          31. Graphics selection
2. Storage utilization     12. Control units         32. History data files*
3. Reserved                13. I/O device load*      33. Benchmark displays*
4. Priv. operations        14. CP owned disks*       34. Correlation coeff.
5. System counters         15. Cache extend. func.*  35. System summary*
6. CP IUCV services        16. DASD I/O assist       36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*   37. CP communications*
8. LPAR data               18. I/O prior. queueing*  38. DASD load
9. Shared segments         19. I/O configuration     39. Minidisk cache*
A. Shared data spaces      1A. I/O config. changes   3A. Storage mgmt. data*
B. Virt. disks in stor.                              3B. Proc. load & config
C. Transact. statistics    User Data                 3C. Logical part. load
D. Monitor data            21. User resource usage*  3D. Response time (all)
E. Monitor settings        22. User paging load*     3E. RSK data menu*
F. System settings         23. User wait states*     3F. Scheduler queues
G. System configuration    24. User response time*   3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*  3H. SFS/BFS logs menu*
                           26. User communication*   3I. System log
I. Exceptions              27. Multitasking users*   3K. TCP/IP data menu*
```

Figure 12-3 Performance Toolkit 3270 Interface Main Menu window

Drilling down into report panels

You should now be able to use the active report panels. To drill down into these, move the cursor to any of the titles that are active (active titles display the number or letter in white, inactive titles in green). Some of the more useful report panels to drill down into are:

```
21. User resource usage
22. User paging load
23. User wait states
28. User configuration
29. Linux systems
33. Benchmark displays
```

For example, to drill down into the Benchmark submenu panel, enter the following command:

```
Command ===> 33
```

Then type `S` over the period on the left side of the submenu panel in the row corresponding to the report you wish to see.

12.3 Monitoring Linux

Measurements can show resource consumption of the Linux guest as measured and dispatched by the VM host. It is also possible to measure performance data from within the Linux guest itself. To monitor Linux performance data at this level, a data gatherer process must be running within each Linux guest you wish to monitor. There are different ways of gathering this data. We recommend that data be gathered in the kernel. All modern Linux distributions have been enabled for the kernel to gather performance data.

12.3.1 Monitoring Linux performance data from the kernel

To monitor Linux performance data directly from the kernel, the following must be true:

- The `APPLMON` option must be set in the user directory.
- Applmon data monitoring must be built into the kernel.

The first requirement should be true because the `OPTION APPLMON` was set for the cloner, the golden image, and for Linux user IDs in earlier sections.

For the second requirement, details of this function are described in Chapter 15, *Linux monitor stream support for z/VM* in the manual *Device Drivers, Features, and Commands* on the web at:

`http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html`

A quick description of how to use this built-in monitoring function follows.

- Start an SSH session to a Linux system. In this example, `LINUX01` is used.
- 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 with the **`modprobe`** command and verify that 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        1844  0
appldata_os             2987  0
appldata_mem            1966  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:

`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 processor 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 may 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 Viewing Linux data in the Performance Toolkit

After the system has had some time to collect data, you should be able to use the Performance Toolkit to view Linux performance data. To view that data, drill down into menu 29, `Linux systems`. This can be done either from the browser interface or the 3270 interface as shown in Figure 12-4.

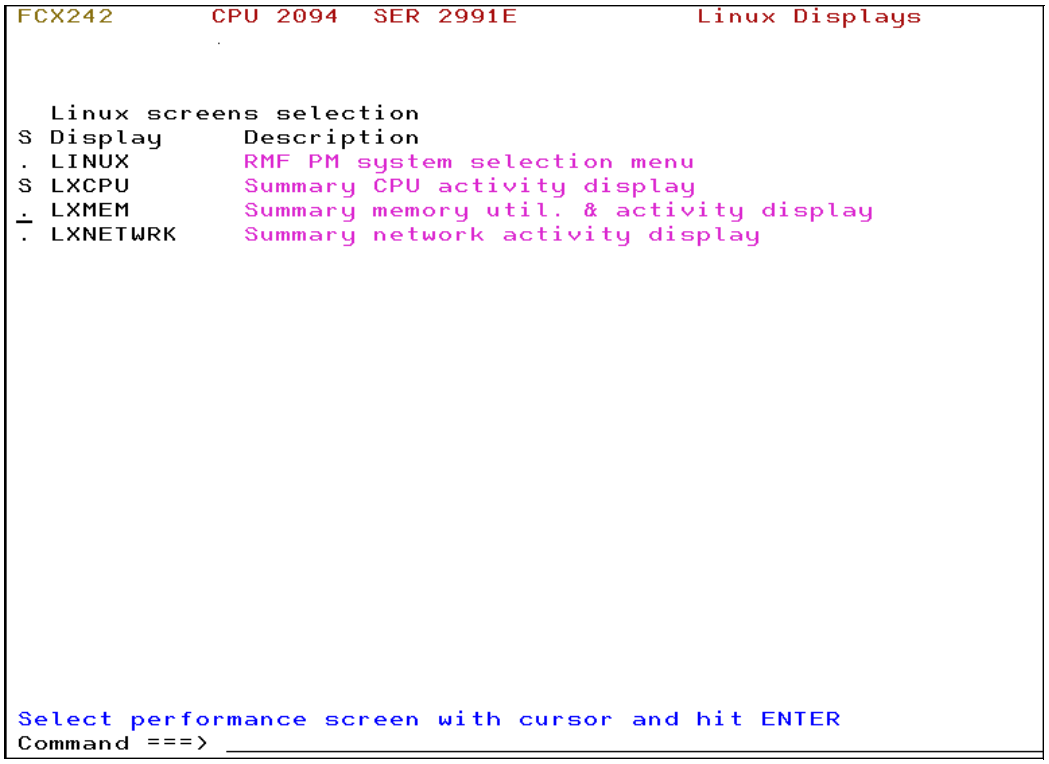


Figure 12-4 Linux Guest Systems submenu

Then type `S` over the period on the left side of the submenu panel in the row corresponding to the report you wish to see. You should see a new report panel with the Linux guest systems processor overview.

You can also use a web interface to view the same data. You would drill down into menu 29 of Linux systems and you should see the drill-down LXCPU (Linux CPU), LXMEM (Linux memory) and LXNET (Linux Network) links hot.

A

Important z/VM files

z/VM differs from Linux concerning 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 12-1 provides a summary and the location of important z/VM configuration files.

Table 12-1 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. If DirMaint is running, the current `USER DIRECT` file can be obtained with the command **`DIRM USER WITHPASS`**.
`PROFILE TCPIP` or `<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.1 Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors

A.1.1 XEDIT cheat sheet

XEDIT has line commands that are typed on the command line (===>), and prefix commands that are typed over the line numbers on the left side of the panel.

Line commands

```
a                     Add a line
a<n>                  Add ‘n’ lines
c/<old>/<new>/ <n> <m>Search for string ‘old’ and replace it with ‘new’ for ‘n’ lines
    below the current line and ‘m’ times on each line. ‘*’ can be used for ‘n’ and ‘m’
/<string>             Search for ‘string’ from the current line
-/<string>            Search backwards for ‘string’
all /<string>/        Show all occurences of ‘string’ and hide other lines
bottom                Move to the bottom of the file
top                   Move to the top of the file
down <n>              Move down ‘n’ lines
up <n>                Move up ‘n’ lines
file                  Save the current file and exit XEDIT
ffile                 Save the current file and exit but don’t warn of overwrite
save                  Save the current file but don’t exit
quit                  Exit XEDIT if no changes have been made
qquit                 Exit XEIDT even if changes have not been saved
left <n>              Shift ‘n’ characters to the left
right <n>             Shift ‘n’ characters to the right
get <file>            Copy file and insert past the current line
:<n>                  Move to line ‘n’
?                     Display last command
=                     Execute last command
x <file>              Edit ‘file’ and put it into the XEDIT “ring”
x                     Move to the next file in the ring
```

Prefix commands

```
a     Add one line
a<n>  Add 'n' lines
c     Copies one line
cc    Copies a block of lines
d     Deletes one line
dd    Deletes a block of lines
f     Line after which a copy (c) or a move (m) is to be inserted
p     Line before which a copy (c) or a move (m) is to be inserted
i     Insert a line
i<n>  Insert 'n' lines
m     Move one line
mm    Move a block of lines
"     Replicate a line
"<n>  Replicate a line 'n' times
""    Replicate a block of lines
```

A.1.2 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:

```
i      brings you back to input mode
dd     deletes a line and puts it in the buffer
<n>dd  delete <n> lines
x      delete a character
dw     delete a word
p      add the buffer past the current location
P      add the buffer before the current location
o      add a line and go into insert mode
/string - search for string
n      do the last command again (this can be powerful)
jkl;   cursor movement
A      add text at the end of the line
<nn>G  go to line <nn>
G      go to the last line in the file
yy     yank a line (copy into buffer)
<n>yy  yank n lines
```

3. Command line mode - pressing the colon **:** key brings you to this mode:

```
:wq       save (write & quit)
:q!       quit and discard changes
:<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
```

B

z/VM and Linux code

This section lists z/VM REXX EXECs and a macro, as well as some Linux code.

B.1 z/VM REXX EXECs and XEDIT macros

This section lists three z/VM REXX EXECs, one XEDIT macro and a sample parameter file.

B.1.1 The CPFORMAT EXEC

The following is the code for the EXEC that formats multiple disks using **CPFMTXA** (described in 4.6.1, “Formatting the paging volumes” on page 50):

```
/*+---------------------------------------------------------------+*/
/*| EXEC: CPFORMAT - wrapper around CPFMTXA to format many DASD    |*/
/*|  retVal: 0 - success                                          |*/
/*|          1 - help was asked for or given                      |*/
/*|          2 - user is not sure                                 |*/
/*|          3 - DASD (minidisk) range is not valid               |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT|*/
/*+---------------------------------------------------------------+*/
/* For details on how this EXEC is used, see one of the two  books:
"z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247931.html
 -or-
"z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 6"
on the Web at: http://www.redbooks.ibm.com/abstracts/SG247932.html */

/*-----------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
-----------------------------------------------------------------*/
firstChar = 'U' /* change this for an LPAR ID other than 'U' */
parse upper arg dasds "AS " type
if ((dasds = '') | (dasds = '?')) then call help
labelPrefix = getLabelPrefix(firstChar type)
numDasd = parseDasd(dasds)
answer = areYouSure(type)
if (answer = 'Y') then  /* the user is sure */
do
  formatted = ""
  retVal = doFormat(labelPrefix numDasd type)
  call doReport retVal
end
else
  retVal = 2
exit retVal

/*+---------------------------------------------------------------+*/
```

```
help: procedure expose firstChar
/*+---------------------------------------------------------------------+*/
  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 '||firstChar||'<t><xxxx> where:'
  say '    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)'
  say '    <xxxx> is the 4 digit address'
  say ''
  say 'Syntax is:'
  say "                                              .-PAGE-."
  say "   >>--CPFORMAT--.-rdev--------------.--AS---+-PERM-+--------><"
  say "                 | <---------------< |       '-SPOL-'"
  say "                 '-rdev1-rdev2-------'  "
  say ''
exit 1

/*+---------------------------------------------------------------------+*/
areYouSure: procedure
/*| Show minidisks, ask are you sure                                    |*/
/*|  parm 1: type - PERM, PAGE, or SPOL                                 |*/
/*|  retVal: firstChar - LPAR identifier, 'V' by default                |*/
/*+---------------------------------------------------------------------+*/
  arg type
  say ''
  say 'WARNING - this will destroy data!'
  say 'ARE YOU SURE you want to format the DASD as' type 'space (y/n)?'
  parse upper pull answer
return  substr(answer, 1, 1) /* from areYouSure */

/*+---------------------------------------------------------------------+*/
getLabelPrefix: procedure
/*| Return first two chararcters of label                               |*/
/*|  parm 1: firstChar - LPAR identifier, 'V' by default                |*/
/*|  retVal: the two character label prefix                             |*/
/*+---------------------------------------------------------------------+*/
  arg firstChar type
  select
  when (type = PERM) then
    labelPrefix = firstChar||'M' /* for VM Minidisk */
  when (type = PAGE) then
    labelPrefix = firstChar||'P' /* for VM Page */
  when (type = SPOL) then
    labelPrefix = firstChar||'S' /* for VM Spool */
  otherwise
    do
      say 'Error: "AS" must be present, type must be PERM, PAGE or SPOL'
      call help
    end /* otherwise */
  end /* select */
return labelPrefix /* from getLabelPrefix */

/*+---------------------------------------------------------------------+*/
```

```
parseDasd: procedure expose dasdList.
/*| parse all dasd into an array verifying all are attached          |*/
/*|  parm 1: dasds - the list of dasd passed in                      |*/
/*|  retVal: number of DASD in dasdList                              |*/
/*+------------------------------------------------------------------+*/
  arg dasds
  numDasd = 0
  say ''
  say 'Format the following DASD:'
  do while (dasds <> '')
    parse upper var dasds dasd dasds
    dashPos = pos('-', dasd)
    if (dashPos = 0) then  /* there is just one DASD */
    do
      numDasd = numDasd + 1
      dasdList.numDasd = dasd
      'CP Q MDISK' dasdList.numDasd 'LOCATION'
      if (rc <> 0) then
      do
        say 'Return code from Q MDISK =' rc
        say 'Are all DASD ATTached?'
        exit 3
      end
      call checkReserved(dasdList.numDasd)
    end /* do */
    else /* process the range of DASD */
    do
      startRange = substr(dasd, 1, dashPos - 1)
      endRange = substr(dasd, dashPos + 1, length(dasd) - dashPos)
      do i = x2d(startRange) to x2d(endRange)
        numDasd = numDasd + 1
        dasdList.numDasd = d2x(i)
        'CP Q MDISK' dasdList.numDasd 'LOCATION'
        if (rc <> 0) then
        do
          say 'Return code from Q MDISK =' rc
          exit 3
        end
        call checkReserved(dasdList.numDasd)
      end /* do i */
    end /* else */
  end /* do while */
return numDasd /* from parseDasd */

/*+------------------------------------------------------------------+*/
doFormat: procedure expose dasdList. formatted
/*| Format all DASD specified using CPFMTXA                          |*/
/*|  parm 1: labelPrefix - the two character label prefix            |*/
/*|  parm 2: numDasd - number of DASD in the array dasdList          |*/
/*|  parm 3: type - the type of DASD format                          |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  arg labelPrefix numDasd type
  'CP TERM MORE 1 1'
  do i = 1 to numDasd
```

```
    label = getLabel(labelPrefix dasdList.i)
    retVal = formatOne(dasdList.i type label)
    if (retVal ^= 0) then
    do
      say "Error from CPFMTXA on DASD" label "rc =" retVal
      leave /* error - abort! */
    end
    formatted = formatted label
  end /* do i = */
  'CP TERM MORE 50 10'
return retVal /* from doFormat */

/*+---------------------------------------------------------------+*/
checkReserved: procedure
/*| Try copying an already formatted DASD then relabelling it     |*/
/*|  parm 1: source                                               |*/
/*|  parm 2: target                                               |*/
/*|  parm 3: label                                                |*/
/*+---------------------------------------------------------------+*/
  arg dasd
  /* create a list of reserved dasd - this is somewhat hokey to be sure
     but it's better to be hokey than to format system minidisks! */
  resvd1 = "0122 0123 0124 0125 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 formatted
/*| Report on the newly labelled DASD                             |*/
/*|  parm 1: formatSuccess - 0=all is well, non-0= a format failed|*/
/*|  retVal: 0 = success                                          |*/
/*+---------------------------------------------------------------+*/
  arg formatSuccess
  if (formatSuccess ^= 0) then
    say 'Error was encountered! retVal from CPFMTXA =' formatSuccess
    if (formatted = '') then
      say "No DASD were successfully formatted"
    else
      say "DASD successfully formatted:" formatted
  'DETACH' dasds
  'ATTACH' dasds '*'
  say ''
  say 'DASD status after:'
```

```
  'CP Q MDISK' dasds 'LOCATION'
return 0 /* from doReport */

/*+---------------------------------------------------------------+*/
formatOne: procedure
/*| Format a DASD via DDR                                         |*/
/*|  parm 1: disk - the vaddr to be formatted                     |*/
/*|  parm 2: type - PAGE, SPOL or PERM                            |*/
/*|  parm 3: label - the six character label                      |*/
/*+---------------------------------------------------------------+*/
  arg disk type label
  queue 'FORMAT'
  queue disk
  queue '0 END'
  queue label
  queue 'YES'
  queue type '0 END'
  queue 'END'
  'CPFMTXA'
  retVal = rc
return retVal /* from formatOne */

/*+---------------------------------------------------------------+*/
getLabel: procedure
/*| Compose the six character label of a minidisk                 |*/
/*|  parm 1: labelPrefix - first two characters of label          |*/
/*|  parm 2: disk - vaddr of length 1, 2, 3 or 4                  |*/
/*|  return: the 6 character label                                |*/
/*+---------------------------------------------------------------+*/
  arg labelPrefix disk
  diskLen = length(disk)
  select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
  end /* select */
return label /* from getLabel */
```

B.1.2 The CHPW610 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 6.1 `USER DIRECT` file:

```
/*+---------------------------------------------------------------+*/
/* CHPW610 XEDIT - change all passwords in z/VM 6.1 USER DIRECT file */
/*+---------------------------------------------------------------+*/

/*---------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
```

```
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
---------------------------------------------------------------------*/
parse arg fn ft fm '(' options ')' newPass .
if (length(newPass) > 8) then
do
  say "Error: new password must be 8 characters or fewer"
  exit
end
say ''
say 'Changing all passwords to:' newPass
say ''

/* set some values */
'command set stay on'
'command set num on'
'command set nulls on'
'command set serial off'
'command set cmdline bottom'
'command set curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'
'command top'


/* change user ID passwords */
'command c/USER MAINT MAINT/USER MAINT' newPass'/*'
'command c/USER AVSVM AVSVM/USER AVSVM' newPass'/*'
'command c/USER TSAFVM TSAFVM/USER TSAFVM' newPass'/*'
'command c/USER GCS GCS/USER GCS' newPass'/*'
'command c/USER GCSXA GCSXA/USER GCSXA' newPass'/*'
'command c/USER AUDITOR AUDITOR/USER AUDITOR' newPass'/*'
'command c/USER AUTOLOG1 AUTOLOG1/USER AUTOLOG1' newPass'/*'
'command c/USER AUTOLOG2 AUTOLOG2/USER AUTOLOG2' newPass'/*'
'command c/USER BLDCMS  BLDCMS/USER BLDCMS' newPass'/*'
'command c/USER BLDNUC  BLDNUC/USER BLDNUC' newPass'/*'
'command c/USER BLDRACF  BLDRACF/USER BLDRACF' newPass'/*'
'command c/USER BLDSEG BLDSEG/USER BLDSEG' newPass'/*'
'command c/USER CMS1  CMS1/USER CMS1' newPass'/*'
'command c/USER CMSBATCH CMSBATCH/USER CMSBATCH' newPass'/*'
'command c/USER DISKACNT DISKACNT/USER DISKACNT' newPass'/*'
'command c/USER EREP EREP/USER EREP' newPass'/*'
'command c/USER IBMUSER IBMUSER/USER IBMUSER' newPass'/*'
'command c/USER LGLOPR  LGLOPR/USER LGLOPR' newPass'/*'
'command c/USER MIGMAINT MIGMAINT/USER MIGMAINT' newPass'/*'
'command c/USER MONWRITE MONWRITE/USER MONWRITE' newPass'/*'
'command c/USER OP1 OP1/USER OP1' newPass'/*'
'command c/USER OPERATNS OPERATNS/USER OPERATNS' newPass'/*'
'command c/USER OPERATOR OPERATOR/USER OPERATOR' newPass'/*'
'command c/USER OPERSYMP OPERSYMP/USER OPERSYMP' newPass'/*'
```

```
'command c/USER SYSADMIN SYSADMIN/USER SYSADMIN' newPass'/*'
'command c/USER SYSDUMP1 SYSDUMP1/USER SYSDUMP1' newPass'/*'
'command c/USER SYSMAINT SYSMAINT/USER SYSMAINT' newPass'/*'
'command c/USER SYSMON  SYSMON/USER SYSMON' newPass'/*'
'command c/USER VMRMADMN VMRMADMN/USER VMRMADMN' newPass'/*'
'command c/USER VMRMSVM VMRMSVM/USER VMRMSVM' newPass'/*'
'command c/USER VMSERVR VMSERVR/USER VMSERVR' newPass'/*'
'command c/USER VMSERVS VMSERVS/USER VMSERVS' newPass'/*'
'command c/USER VMSERVU VMSERVU/USER VMSERVU' newPass'/*'
'command c/USER VMUTIL VMUTIL/USER VMUTIL' newPass'/*'
'command c/USER VSMPROXY VSMPROXY/USER VSMPROXY' newPass'/*'
'command c/USER VSMREQIN VSMREQIN/USER VSMREQIN' newPass'/*'
'command c/USER VSMREQIU VSMREQIU/USER VSMREQIU' newPass'/*'
'command c/USER VSMSERVE VSMSERVE/USER VSMSERVE' newPass'/*'
'command c/USER VSMWORK1 VSMWORK1/USER VSMWORK1' newPass'/*'
'command c/USER VSMWORK2 VSMWORK2/USER VSMWORK2' newPass'/*'
'command c/USER VSMWORK3 VSMWORK3/USER VSMWORK3' newPass'/*'
'command c/USER ZVMMAPLX MAINT/USER ZVMMAPLX' newPass'/*'
'command c/USER 5684042J 5684042J/USER 5684042J' newPass'/*'
'command c/USER 4OSASF4O 4OSASF4O/USER 4OSASF4O' newPass'/*'
'command c/USER OSADMIN1 OSADMIN1/USER OSADMIN1' newPass'/*'
'command c/USER OSADMIN2 OSADMIN2/USER OSADMIN2' newPass'/*'
'command c/USER OSADMIN3 OSADMIN3/USER OSADMIN3' newPass'/*'
'command c/USER OSAMAINT OSAMAINT/USER OSAMAINT' newPass'/*'
'command c/USER OSASF OSASF/USER OSASF' newPass'/*'
'command c/USER 6VMRSC1O 6VMRSC1O/USER 6VMRSC1O' newPass'/*'
'command c/USER RSCS RSCS/USER RSCS' newPass'/*'
'command c/USER RSCSAUTH RSCSAUTH/USER RSCSAUTH' newPass'/*'
'command c/USER RSCSDNS RSCSDNS/USER RSCSDNS' newPass'/*'
'command c/USER XCHANGE XCHANGE/USER XCHANGE' newPass'/*'
'command c/USER 6VMTCP1O 6VMTCP1O/USER 6VMTCP1O' newPass'/*'
'command c/USER TCPIP TCPIP/USER TCPIP' newPass'/*'
'command c/USER TCPMAINT TCPMAINT/USER TCPMAINT' newPass'/*'
'command c/USER ADMSERV ADMSERV/USER ADMSERV' newPass'/*'
'command c/USER DHCPD DHCPD/USER DHCPD' newPass'/*'
'command c/USER DTCVSW1  DTCVSW1/USER DTCVSW1' newPass'/*'
'command c/USER DTCVSW2  DTCVSW2/USER DTCVSW2' newPass'/*'
'command c/USER FTPSERVE FTPSERVE/USER FTPSERVE' newPass'/*'
'command c/USER IMAP IMAP/USER IMAP' newPass'/*'
'command c/USER IMAPAUTH IMAPAUTH/USER IMAPAUTH' newPass'/*'
'command c/USER LDAPSRV  LDAPSRV/USER LDAPSRV' newPass'/*'
'command c/USER LPSERVE LPSERVE/USER LPSERVE' newPass'/*'
'command c/USER MPROUTE MPROUTE/USER MPROUTE' newPass'/*'
'command c/USER NAMESRV NAMESRV/USER NAMESRV' newPass'/*'
'command c/USER NDBPMGR NDBPMGR/USER NDBPMGR' newPass'/*'
'command c/USER NDBSRVO1 NDBSRVO1/USER NDBSRVO1' newPass'/*'
'command c/USER PORTMAP PORTMAP/USER PORTMAP' newPass'/*'
'command c/USER REXECD REXECD/USER REXECD' newPass'/*'
'command c/USER SMTP SMTP/USER SMTP' newPass'/*'
'command c/USER SNALNKA SNALNKA/USER SNALNKA' newPass'/*'
'command c/USER SNMPD SNMPD/USER SNMPD' newPass'/*'
'command c/USER SNMPQE SNMPQE/USER SNMPQE' newPass'/*'
'command c/USER SNMPSUBA SNMPSUBA/USER SNMPSUBA' newPass'/*'
'command c/USER SSLSERV SSLSERV/USER SSLSERV' newPass'/*'
'command c/USER TFTPD TFTPD/USER TFTPD' newPass'/*'
'command c/USER UFTD    UFTD/USER UFTD' newPass'/*'
'command c/USER VMKERB VMKERB/USER VMKERB' newPass'/*'
'command c/USER VMNFS VMNFS/USER VMNFS' newPass'/*'
'command c/USER X25IPI X25IPI/USER X25IPI' newPass'/*'
'command c/USER 6VMDIR1O 6VMDIR1O/USER 6VMDIR1O' newPass'/*'
```

```
'command c/USER 6VMRAC10 6VMRAC10/USER 6VMRAC10' newPass'/*'
'command c/USER RACFSMF RACFSMF/USER RACFSMF' newPass'/*'
'command c/USER RACFVM RACFVM/USER RACFVM' newPass'/*'
'command c/USER RACMAINT RACMAINT/USER RACMAINT' newPass'/*'
'command c/USER 6VMPTK10 6VMPTK10/USER 6VMPTK10' newPass'/*'
'command c/USER PERFSVM  PERFSVM/USER PERFSVM' newPass'/*'
'command c/USER 5VMHCD40 5VMHCD40/USER 5VMHCD40' newPass'/*'
'command c/USER CBDIODSP CBDIODSP/USER CBDIODSP' newPass'/*'
'command c/USER GSKADMIN GSKADMIN/USER GSKADMIN' newPass'/*'
'command c/USER LNXMAINT LNXMAINT/USER LNXMAINT' newPass'/*'

/* change mindisk passwords */
'command c/ALL      WRITE    MULTIPLE/ALL' newPass newPass'/*'
'command c/RADMSERV WADMSERV MADMSERV/'newPass newPass newPass'/*'
'command c/RAUDITOR WAUDITOR MAUDITOR/'newPass newPass newPass'/*'
'command c/RAUTOLOG WAUTOLOG MAUTOLOG/'newPass newPass newPass'/*'
'command c/RAVSOBJ  WAVSOBJ  MAVSOBJ/'newPass newPass newPass'/*'
'command c/RBATCH   WBATCH   MBATCH/'newPass newPass newPass'/*'
'command c/RCATALOG WCATALOG/'newPass newPass'/*'
'command c/RCONTROL WCONTROL/'newPass newPass'/*'
'command c/RCRRLOG1 WCRRLOG1/'newPass newPass'/*'
'command c/RCRRLOG2 WCRRLOG2/'newPass newPass'/*'
'command c/RDATA    WDATA/'newPass newPass'/*'
'command c/RDHCPD   WDHCPD   MDHCPD/'newPass newPass newPass'/*'
'command c/RDTCVSW1 WDTCVSW1 MDTCVSW1/'newPass newPass newPass'/*'
'command c/RDTCVSW2 WDTCVSW2 MDTCVSW2/'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/RGSKADMN WGSKADMN MGSKADMN/'newPass newPass newPass'/*'
'command c/RIMAP    WIMAP    MIMAP/'newPass newPass newPass'/*'
'command c/RLDAPSRV WLDAPSRV MLDAPSRV/'newPass newPass newPass'/*'
'command c/RLOG1    WLOG1/'newPass newPass'/*'
'command c/RLOG2    WLOG2/'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/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/RSNMPSUB WSNMPSUB MSNMPSUB/'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/ALL      WTCPMAIN MTCPMAIN/ALL' newPass newPass'/*'
'command c/MR READ/'MR newPass'/*'
```

B.1.3 PROFILE EXEC for Linux user IDs

This section lists the code for the PROFILE EXEC that is shared among Linux user IDs from the LNXMAINT 192 disk:

```
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 300 524288'  /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else  /* user is interactive -> prompt */
 do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
 end /* else */
```

B.2 Linux code

This section lists the code for the /sbin/clone.sh script that clones from the golden Linux image to a target virtual machine:

```
#!/bin/sh
#
# clone.sh <LinuxUserID> - clone a Linux server running under z/VM
#
# For details on how this script works see the book:
#  "z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1"
#  on the Web at: http://www.linuxvm.org/present
#
# ------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# ------------------------------------------------------------------------

#+-------------------------------------------------------------------------+
function help()
# give help
#+-------------------------------------------------------------------------+
```

```
 {
  echo "Usage: clone [options] from <sourceID> to <targetID>"
  echo ""
  echo "  Clone Linux from sourceID 100 and 101 minidisks to targetID"
  echo "  options:"
  echo "    -v or --verbose: verbose"
  echo ""
  echo "Example: clone.sh from s11gold to linux01"
  exit 1
 }

#+-----------------------------------------------------------------------------+
function processArguments()
# Parse command line arguments
# Args: The arguments passed in to the script
#+-----------------------------------------------------------------------------+
 {
  verbose="off"
  sourceID="none"
  targetID="none"
  while (( "$#" )); do
    case $1 in
      -v|--verbose)
         verbose="on"
         ;;
      from)
        shift
        sourceID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold source ID to upper case
        ;;
      to)
        shift
        targetID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold target ID to upper case
        ;;
    esac
    shift
  done
  if [ $sourceID = "none" ]; then # source user ID was not passed
    echo "Error: Source Linux user ID not supplied"
    help
  fi
  if [ $targetID = "none" ]; then # target user ID was not passed
    echo "Error: Target Linux user ID not supplied"
    help
  fi
 }

#+-----------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+-----------------------------------------------------------------------------+
 {
  echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
  return $retVal
```

```
 }

#+-------------------------------------------------------------------------+
function checkID()
# Verify user ID exists and is logged off
#    Arg 1: The user ID to check
#+-------------------------------------------------------------------------+
 {
  userID=$1
  echo "Checking that $userID exists and is not logged on ..."
  CPcmd QUERY $userID
  rc=$?
  case $rc in
    0)  # user ID is logged on or disconnected
      echo "$userID user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$userID user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "Return code of $rc unexpected from QUERY $userID"
      echo "User ID must exist and be logged off"
      exit 4
  esac
 }

#+-------------------------------------------------------------------------+
function prepareIPaddr()
# Set the variable "newIPaddr" by adding a backslash before any "."s
#    Arg 1: The IP address to be modified
#+-------------------------------------------------------------------------+
 {
  newIPaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }

#+-------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+-------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+-------------------------------------------------------------------------+
function copyDisk()
# Use FLASHCOPY to copy a disk, if it fails, fall back to dasdfmt then dd
```

```
#   Arg 1: Source vaddr
#   Arg 2: Target vaddr
#+----------------------------------------------------------------------------+
 {
  source=$1
  target=$2
  echo ""
  echo "FLASHCOPYing $source to $target ..."
  CPcmd FLASHCOPY $source 0 end to $target 0 end
  if [ $? != 0 ]; then
    echo "FLASHCOPY failed, falling back to dasdfmt and dd ..."
    chccwdev -e $source
    if [ $? != 0 ]; then exit 7; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 8; fi
    sleep 1
    srcDev=/dev/$(egrep ^0.0.$source /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 5; fi
    tgtDev=/dev/$(egrep ^0.0.$target /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 6; fi
    echo "dasdfmt-ing $tgtDev ..."
    dasdfmt -y -b 4096 -f $tgtDev
    if [ "$?" != 0 ]; then exit 9; fi
    echo "dd-ing $srcDev to $tgtDev ..."
    dd bs=4096 if=$srcDev of=$tgtDev oflag=sync
    if [ "$?" != 0 ]; then exit 10; fi
    sync
    echo "disabling and re-enabling $target ..."
    chccwdev -d $target
    if [ $? != 0 ]; then exit 11; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 12; fi
    sync
  fi
 }

#+----------------------------------------------------------------------------+
function getNetworkInfo()
# Bring 191 minidisk online to check for two parameter files
#+----------------------------------------------------------------------------+
 {
  # recycle 191 to pick up latest changes
  chccwdev -d 191
  chccwdev -e 191
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "unable to enable 191, rc from chccwdev = $rc"
    exit 13
  fi
  udevadm settle
  CMSdisk=`lsdasd | grep 0191 | awk '{ print $3 }'`
  cmsfslst -d /dev/$CMSdisk | grep -i $sourceID | grep PARM-S11
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $sourceID PARM-S11 not found on 191 minidisk. Exiting"
    exit 14
  fi
  cmsfslst -d /dev/$CMSdisk | grep -i $targetID | grep PARM-S11
  rc=$?
  if [ $rc != 0 ]; then
```

```
    echo "Error: $targetID PARM-S11 not found on 191 minidisk. Exiting"
    exit 15
  fi

# get informaton about target
  export local $(cmsfscat -a -d /dev/$CMSdisk $targetID.PARM-S11)
  tagetHostname=$Hostname
  targetIP=$HostIP
  targetDNS=$Nameserver
  targetGW=$Gateway
  targetMask=$Netmask
  targetBroadcast=$Broadcast
  prepareVaddr $ReadChannel
  targetReaddev=$newVaddr
  prepareVaddr $WriteChannel
  targetWritedev=$newVaddr
  prepareVaddr $DataChannel
  targetDatadev=$newVaddr

# get information about source
  export local $(cmsfscat -a -d /dev/$CMSdisk $sourceID.PARM-S11)
  sourceHostname=$Hostname
  prepareIPaddr $HostIP
  sourceIP=$newIPaddr
  prepareIPaddr $Nameserver
  sourceDNS=$newIPaddr
  prepareIPaddr $Gateway
  sourceGW=$newIPaddr
  prepareIPaddr $Netmask
  sourceMask=$newIPaddr
  prepareIPaddr $Broadcast
  sourceBroadcast=$newIPaddr
  prepareVaddr $ReadChannel
  sourceReaddev=$newVaddr
  prepareVaddr $WriteChannel
  sourceWritedev=$newVaddr
  prepareVaddr $DataChannel
  sourceDatadev=$newVaddr
 }

#+-------------------------------------------------------------------------+
function askAreYouSure()
# Ask "Are you sure?" - if not, then exit
#+-------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: Mindisks 100 and 101 will be copied to $targetID"
  echo "New host name will be:  $tagetHostname"
  echo "New IP address will be: $targetIP"
  echo "Other network data is retrieved from $targetID PARM-S11 on 191 disk"
  echo -n "Are you sure you want to overwrite these disks (y/n): "
  read ans
  if [ $ans != "y" ]; then
    echo "Aborting clone per user input"
    exit 16
  fi
 }

#+-------------------------------------------------------------------------+
function copySystem()
```

```
# For each of two mindisks 100 and 101:
#   -) Link disk
#   -) Enable disk
#   -) Copy disk
#+-----------------------------------------------------------------------------+
 {
  echo "Linking source and target 100 disks ..."
  CPcmd detach 1100
  CPcmd link $sourceID 100 1100 rr
  if [ $? != 0 ]; then exit 17; fi
  CPcmd detach 2100
  CPcmd link $targetID 100 2100 mr
  if [ $? != 0 ]; then exit 18; fi
  echo "Copying 100 disks ..."
  copyDisk 1100 2100
  echo "Take 1100 Offline...."
  chccwdev -d 1100
  CPcmd det 1100
  # don't detach 2100 yet because it has to be modified

  echo " "
  echo "---------------------------------------"
  echo "Linking source and target 101 disks ..."
  CPcmd detach 1101
  CPcmd link $sourceID 101 1101 rr
  if [ $? != 0 ]; then exit 19; fi
  CPcmd detach 2101
  CPcmd link $targetID 101 2101 mr
  if [ $? != 0 ]; then exit 20; fi
  echo "Copying 101 disks ..."
  copyDisk 1101 2101
  echo "Taking 1101 Offline..."
  chccwdev -d 1101
  CPcmd det 1101
  echo "Taking 2101 Offline..."
  chccwdev -d 2101
  CPcmd det 2101
 }

#+-----------------------------------------------------------------------------+
function modifyClone()
# Mount newly copied system and modify networking info
#+-----------------------------------------------------------------------------+
 {
  targetVaddr=2100
  targetDir="/mnt/targetLinux"         # directory of target Linux system
  echo "Mounting newly cloned image over $targetDir ..."
  if [ ! -d $targetDir ]; then
    mkdir $targetDir
    rc=$?
    if [ $rc != 0 ]; then
      echo "Error: mkdir $targetDir failed with $rc"
      exit 21
    fi
  fi
  # recycle target mount point
#  chccwdev -d $targetVaddr
  chccwdev -e $targetVaddr
  rc=$?
  if [ $rc != 0 ]; then
```

```
    echo "Fatal error: chccwdev -e $targetVaddr failed with $rc"
    CPcmd DET $targetVaddr
    exit 22
  fi
  sleep 2
  tDev=/dev/$(egrep ^0.0.$targetVaddr /proc/dasd/devices | awk '{ print $7 }')1
    if [ "$?" != 0 ]; then exit 23; fi
  echo "Mounting $tDev over $targetDir ..."
  mount $tDev $targetDir
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: 'mount $tDev $targetDir' failed with $rc"
    lsdasd
    CPcmd DET $targetVaddr
    exit 24
  fi
  echo "Modifying cloned image under $targetDir ..."
  sed --in-place -e "s/$sourceHostname/$tagetHostname/g" \
    $targetDir/etc/HOSTNAME
  sed --in-place -e "s/$sourceIP/$targetIP/g" \
    -e "s/${sourceHostname%%.*}/${tagetHostname%%.*}/g" $targetDir/etc/hosts
  sed --in-place -e "s/$sourceGW/$targetGW/g" \
    $targetDir/etc/sysconfig/network/routes
  sed --in-place -e "s/$sourceIP/$targetIP/g" \
    -e "s/$sourceBroadcast/$targetBroadcast/g" \
    $targetDir/etc/sysconfig/network/ifcfg-eth0
  sed --in-place -e "s/$sourceDNS/$targetDNS/g" $targetDir/etc/resolv.conf

# Delete SSH keys - sshd will recreate them at first boot
  echo "Removing SSH keys"
  rm $targetDir/etc/ssh/ssh_host*

# Remove any old entry, then copy clone's public key to known_hosts file
  echo "Removing $targetIP from known_hosts file"
  cd /root/.ssh
  grep -v $targetIP known_hosts > known_hosts.temp
  mv known_hosts.temp known_hosts

# clean up
  sleep 1
  sync # sync disks
  umount $targetDir
  sync
  chccwdev -d $targetVaddr
  CPcmd DETACH $targetVaddr
  return 0
 }

# main()
processArguments $@                       # process arguments passed by user
if [ $verbose = "on" ]; then set -vx; fi  # turn on debug
checkID $sourceID                         # user ID must exist and be logged off
checkID $targetID                         # user ID must exist and be logged off
getNetworkInfo                            # get info from parm files
askAreYouSure                             # confirm disks will be overwritten
copySystem                                # copy source disks to target
modifyClone                               # modify newly copied system
CPcmd XAUTOLOG $targetID                  # bring new clone to life
if [ $verbose = "on" ]; then set +vx; fi  # turn off debug
echo "Successfully cloned $sourceID to $targetID"
```

```
echo "You should be able to ping $targetIP within one minute"
exit 0
```

C

Additional material

This appendix refers to additional material that can be downloaded from the Internet.

Locating the web material

The web material associated with this book is available in softcopy on the Internet from the IBM Redbooks web server. Point your web browser at:

ftp://www.redbooks.ibm.com/redbooks/SG247931

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247931.

Using the web material

The additional web material that accompanies this book includes the following files:

File name	*Description*
SG247931.tgz	Directory virt-cookbook-S11SP1/ exists containing: vm/ A directory with z/VM files vm/chpw610.xeditXEDIT macro to change passwords in z/VM 6.1 USER DIRECT vm/cpformat.execREXX EXEC to format many disks using CPFMTXA vm/profile.exec Sample Linux user ID PROFILE EXEC to IPL Linux vm/sample.parm-s11Sample SLES 11 SP1 parameter file vm/sles11s1.execREXX EXEC to kick off SLES 11 SP1 install vm/swapgen.execREXX EXEC to create a VDISK swap space clone.shA cloning script disclaimer.txt Legal disclaimer

Downloading and extracting the web material

Create a subdirectory (folder) on your workstation, and extract the contents of the web material `.tgz` file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *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

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

`ibm.com/redbooks`

Other publications

These publications are also relevant as further information sources:

- *SUSE LINUX Enterprise Server* documentation - on CD1 or DVD1 in the `docu/` directory:
  - *SUSE LINUX Enterprise Server: Installation and Administration*
  - *SUSE LINUX Enterprise Server: Architecture-specific Information*
  - *SUSE LINUX Enterprise Server: Start-up Guide*
- z/VM documentation:

  `http://www.vm.ibm.com/library/`
  - *z/VM Guide for Automated Installation and Service*
  - *z/VM CP Messages and Codes*
  - *z/VM TCP/IP Messages and Codes*
  - *z/VM CP Commands and Utilities Reference*
  - *z/VM CP Planning and Administration*
  - *z/VM Getting Started with Linux on System z9 and zSeries*
  - *z/VM TCP/IP Planning and Customization*
  - *z/VM Performance Toolkit Guide*, SC24-6156
  - *z/VM Performance Toolkit Reference*, SC24-6157
  - The Program Directory for Performance Toolkit for VM

Online resources

These web sites are also relevant as further information sources:

- Documentation for System z Linux Development stream:
  `http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html`
- Linux for zSeries and S/390 Device Drivers, Features, and Commands, LNUX-1403:
  `http://public.dhe.ibm.com/software/dw/linux390/docu/lk36dd08.pdf`
- The Linux for zSeries and S/390 portal:
  `http://linuxvm.org/`
- The linux-390 list server:
  `http://www2.marist.edu/htbin/wlvindex?linux-390`
- Linux on System z and S/390 developerWorks:
  `http://awlinux1.alphaworks.ibm.com/developerworks/linux390/index.shtml`
- z/VM publications:
  `http://www.vm.ibm.com/pubs/`
- z/VM performance tips:
  `http://www.vm.ibm.com/perf/tips/`

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

z/VM and Linux on IBM System z
The Virtualization Cookbook for SLES 11 SP1

Hands-on instructions for installing z/VM and Linux on the mainframe

Updated information for z/VM 6.1 and Linux SLES 11 SP1

A new, more versatile file system layout

This IBM Redbooks publication describes how to create your own Linux virtual servers on IBM® System z hardware under z/VM. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing and configuring z/VM in an LPAR and then installing and customizing Linux. You need an IBM System z logical partition (LPAR) with associated resources, z/VM 6.1 media, and SLES 11 SP1 Linux for System z.

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

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7931-00 ISBN 0738435074