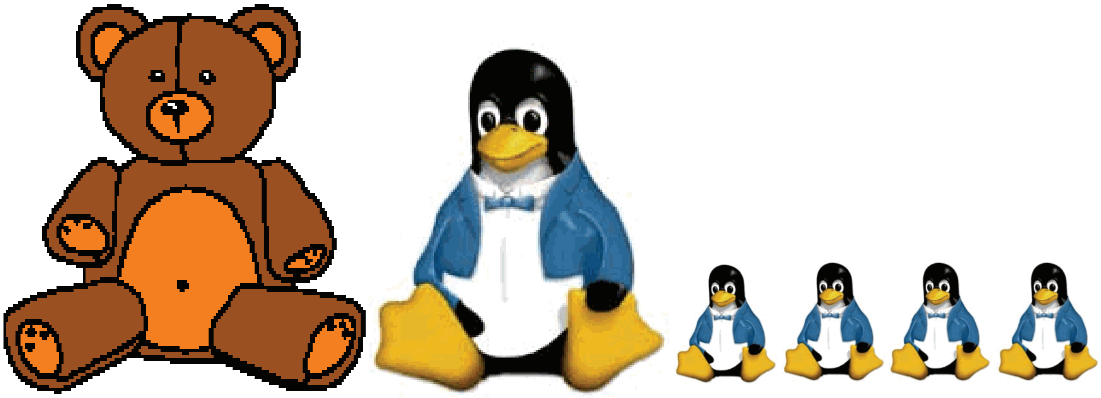
z/VM and Linux on IBM System z:
The Virtualization Cookbook for RHEL 5

A cookbook for installing and customizing z/VM 5.2 and Linux RHEL 5 on the mainframe



Michael MacIsaac, Brad Hinson

February 2007

Contents

Preface

"*The search for truth is more precious than its possession.*"

— Albert Einstein

This book describes how to *roll 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 5.2 media, and a Linux distribution.

In February of 2007, two books were published, 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*

In August of 2006 a book was published that addresses both 31-bit and 64-bit SLES 9:

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

In addition to these books, there are associated REXX EXECs and Linux scripts to help speed up the installation and configuration of both z/VM and Linux. These tools are not IBM products nor formally supported. However, they will be informally supported. They are available on the Web.

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.

Summary of changes to August 2006 version

The second release of the cookbook in August of 2006 was based on the IBM redbook *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9*, SG24-6695-01. The content that was added to that book is as follows:

- Manual cloning: A new section in chapter 9, Cloning a virtual server manually was added so you can better understand the steps involved in the cloning process
- Cloning of IBM Middleware: A new chapter 11 *Cloning IBM Middleware* was added that describes installing, sharing, customizing and cloning IBM WebSphere Application Server, DB2 and MQ Series.
- The **`clone.sh`** script has three new flags added: **`--was`**, **`--db2`** and **`--mqs`** to support the cloning of IBM middleware.

Summary of changes to February 2007 version

The release of two new books in February of 2007 added the following:

- SLES 10 is described in the book *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10.*

- RHEL 5 is described in the book *z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5.*
- Two VDISK swap spaces are used rather than one VDISK and one minidisk - this allows for both more potential swap space and a larger root file system.
- Associated files for the controller are distributed as RPMs - one for SLES 10 and one for RHEL 5. This removes the need to recursively copy the content of the associated tar files.
- A section is added to the chapter “Miscellaneous recipes” entitled “Centralizing home directories for LDAP users”. This ties together a virtual LDAP server, authentication (PAM, NSS), LVM, NFS and automount so that cloned virtual servers share a common user/group/password LDAP database and a common `/home/` directory that is automatically mounted.
- Another section is added to the chapter “Miscellaneous recipes” entitled “Rescuing a Linux system”. This is a quick reference on how to begin to repair a system that is no longer able to boot. Hopefully this section will not be referenced often (or at all :)).
- A Web Application named Data About z/VM and Linux (DAZL) is described in Chapter 13, “Monitoring z/VM and Linux” on page 197. This allows the controller to run a password protected Web Server that makes much z/VM and Linux data quickly available via a browser. It relies on the `/backup/` file system being populated correctly via the **`clone.sh`** script, rsync, key-based authentication and cron.
- The **`clone.sh`** script was moved from `/usr/local/sbin/` to `/sbin/`. The hard coded reference to the master image’s mount point has been removed. Rather, only the root file system minidisk (100) is cloned. That value was made into a variable to more easily allow for multiple master images to be cloned.
- Installing z/VM onto 3390-9s is addressed - a `LBL520-9 EXEC` and `XEDIT` macro are supplied to allow for the relabeling of the system volumes.

The release of two new books also removed some sections:

- The chapter on Cloning of IBM Middleware has been removed - at the time of publishing not all the middleware was not supported on the new distributions.
- DCSS/XIP2 - it does not seem that this environment is being widely used.

Conventions

The following font conventions are used in this book:

`Monospace and bold`	Commands entered by the user on the command line
`<value>`	Value inside angle brackets is to be replaced
`monospace`	File, directory and user ID names

The following command conventions are used in this book:

- z/VM commands are prefixed with **==>**
- z/VM XEDIT subcommands are prefixed with **====>**
- Linux commands running as root are prefixed with **#**
- Linux commands that will not fit on one line are suffixed with \
- Linux commands running as non-root are usually prefixed with **$**

The team that wrote this trilogy

This book was originally written in 2005 by Michael MacIsaac, Jin Xiong and Curtis Gearhart. It was updated in 2006 by Michael MacIsaac, Carlos Ordonez and Jin Xiong. It was updated a third time in late 2006 and early 2007 by Marian Gasparovic taking the lead on the SLES 10 version and Brad Hinson of Red Hat working on the RHEL 5 version. The DCSS/XIP2 section was based on work by Carlos Ordonez, which was in turn based on work by Carsten Otte. Most of the section on the CP INDICATE command was written by Bill Bitner.

Sincere thanks goes out to the following people who contributed to this project in various ways:

Jerry Epker
Fidelity Information Services

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

Carsten Otte, Claudia Prawirakusumah
IBM Germany

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

Bernard Wu
NCCI

Mark Post
Novell

Matthew Eernisse
OpenSource Applications

David Boyes, Adam Thornton
Sine Nomine Associates

Chester Hood
State of Tennessee

Rob van der Heij
Velocity Software

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

Special thanks to Tony Giaccone from IBM Poughkeepsie for always finding another free LPAR and more DASD.

Comments welcome

Your comments are important. You can send your comments in e-mail directly to:

```
mikemac at us.ibm.com
marian.gasparovic at sk.ibm.com
bhinson at redhat.com      // For the RHEL 5 book
```

Notices

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States,

developerWorks®
eServer™
ibm.com®
iSeries™
z/OS®
z/VM®
System z®
AIX®
DirMaint™
Domino®
ECKD™
ETE™
HiperSockets™
Hummingbird®
IBM®
Lotus®
Notes®
Perform™
Requisite®
RACF®
RMF™
S/390®
SLC™
Tivoli®
Workplace™

The following terms are trademarks of other companies:

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

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

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

Part 1

z/VM

This part of the book has an introduction chapter and then discusses z/VM installation, configuration, and servicing.

1

Introduction to z/VM and Linux

"*Everything should be made as simple as possible, but not simpler.*"

— Albert Einstein

Virtualization is hot. The mainframe and z/VM have been doing virtualization for decades. When Linux came to the 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.

With a z/VM and Linux infrastructure, you can reduce the time between deciding on new servers and implementing them because new servers can be deployed in a matter of minutes. This powerful capability can enable you to launch new products and services without planning, purchasing, installing and configuring new hardware and software. Development groups who need test environments built and rebuilt rapidly to efficiently handle change management can also benefit from this unique advantage.

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

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

z/VM 5.2, available in December of 2005, provides major improvements when operating on System z servers with large memory configurations. Scalability is improved with the Control Program (CP) now using memory above 2 GB for a much broader set of operations. Previously, guest pages had to be moved below 2GB 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.

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

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

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

1.1 What is virtualization?

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

1.2 A philosophy adopted in this book

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

1.3 Choices and decisions made in this book

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

- Cloning product versus "roll your own" cloning:

  Cloning products, such as Aduva's Onstage, IBM Tivoli Provisioning Manager, IBM Director function 'z/VM Center' (briefly discussed in "IBM Director and z/VM Center Extension" on page 5) and Levanta, 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 EXECs and scripts in this book.

- Directory Maintenance product versus the `USER DIRECT` file:

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

- Provisioning versus predefined user IDs:

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

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

  Many cloning solutions use an environment which shares the `/usr/` file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. A read-write `/usr/` file system on the virtual servers is chosen to keep things as simple as possible.

- Conventional 3390 ECKD DASD versus FBA disks accessed via SCSI over FCP:

  The System z server has traditionally only supported 3390 (or older 3380) DASD. Support has been extended to include SCSI/FBA disks in storage area networks (SANs). The support of FBA disks is 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:

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

1.4 IBM Director and z/VM Center Extension

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

The z/VM Center extension includes the following tasks:

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

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

1.5 Infrastructure design

To clone Linux, or *provision virtual servers*, there must be a certain infrastructure design in place. A System z server with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-1 on page 6 shows a block diagram of a System z9 with many LPARs. z/VM 5.2 is installed in one of these LPARs. z/VM comes with many user IDs predefined. The most important six IDs are shown in the z/VM LPAR above the dashed line. Below the dashed line, you see the user IDs described in this book. Important z/VM minidisks and configuration files are shown next to each user ID.

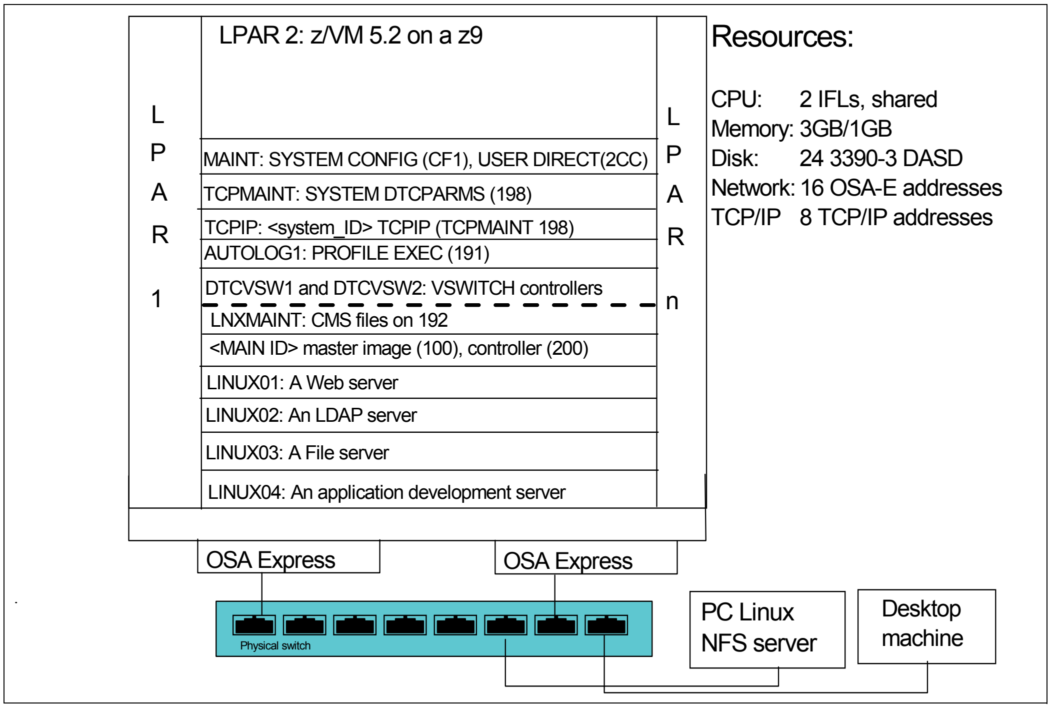

LPAR 2: z/VM 5.2 on a z9
L P A R 1
L P A R n
MAINT: SYSTEM CONFIG (CF1), USER DIRECT(2CC)
TCPMAINT: SYSTEM DTCPARMS (198)
TCPIP: <system_ID> TCPIP (TCPMAINT 198)
AUTOLOG1: PROFILE EXEC (191)
DTCVSW1 and DTCVSW2: VSWITCH controllers
LNXMAINT: CMS files on 192
<MAIN ID> master image (100), controller (200)
LINUX01: A Web server
LINUX02: An LDAP server
LINUX03: A File server
LINUX04: An application development server
OSA Express
OSA Express
Physical switch
PC Linux NFS server
Desktop machine
Resources:
CPU: 2 IFLs, shared
Memory: 3GB/1GB
Disk: 24 3390-3 DASD
Network: 16 OSA-E addresses
TCP/IP 8 TCP/IP addresses

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

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

- **LNXMAINT**: A user ID on which to store files that will be used by both CMS and Linux
- **<MAIN ID>**: The user ID that stores the master (or golden) image that is cloned and IPLed from minidisk 100 and the controller (or cloner) image that does the cloning and is IPLed from minidisk 200. In the RHEL 5 book, this user ID is named `RHEL5` and in the SLES 10 book it is named `SLES10`.
- **LINUX01-04**: The Linux virtual servers described in the *Cloning open source virtual servers* chapter. In addtion, creating `LINUX05-07` is described so you can have additional user IDs to clone to. Each virtual server is configured with a single 3390-3 minidisk which is slightly more than 2 GB of space.

1.6 Usability tests performed for this book

During the writing of this book, many usability tests were conducted. The participants had a variety of skills, but none had both Linux and z/VM system administration skills. By the end of the first day in all of the formal tests, most participants had all completed up to and including Chapter 5, “Servicing z/VM” on page 69, so z/VM was installed, serviced and customized for TCP/IP communications with a highly available VSWITCH. By the end of the second day, most participants had cloned their first Linux virtual server. You should be able to complete most steps in the book in four solid days of work, if all goes well.

1.7 The chapters in this book

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

- Chapter 2, “Planning” on page 9 describes how to plan hardware, software and networking resources. It discusses DASD labeling conventions used in the book and a password planning. Sample worksheets are provided for the examples used in the book, as are blank copies for your use.
- Chapter 3, “Configuring a desktop machine” on page 21 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 29 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 69 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 RHEL 5” on page 83, explains how to set up a temporary NFS server on a Linux PC for the purpose of installing the first two Linux images. After the System z controller Linux is installed, you can copy the Linux install tree to it and retire the Linux PC server.
- Chapter 7, “Installing and configuring RHEL 5” on page 89, describes how to install and configure two Linux images onto the first Linux user ID: the *master image*, which it is cloned from, and the *controller*, which does the cloning among other tasks.
- Chapter 8, “Configuring RHEL 5 for cloning” on page 133 explains how to prepare z/VM user IDs and clone your first virtual server.
- Chapter 9, “Installing Linux with kickstart” on page 149 explains how to use Red Hat’s kickstart tool to install virtual servers.
- Chapter 10, “Servicing Linux with Red Hat Network” on page 155 explains how to use the Red Hat Network and its ability to manage virtual servers.
- Chapter 11, “Cloning open source virtual servers” on page 161, shows how to configure cloned Linux images into the following virtual servers:
  - Web server virtual server
  - LDAP virtual server
  - File and print virtual server
  - Basic application development system
- Chapter 12, “Miscellaneous Recipes” on page 183 describes how to add a logical volume to a Linux system and how to set up a z/VM Discontiguous Saved Segment (DCSS) in conjunction with the Linux eXecute In Place 2 (xip2) file system.
- Chapter 13, “Monitoring z/VM and Linux” on page 197, describes basic steps to begin monitoring z/VM and your new Linux virtual servers.
- Chapter 14, “Backup and restore” on page 223, shows basic steps on how to back up these new systems.
- Appendix A, “References” on page 229, provides references Web sites, books and other pertinent information.
- Appendix B, “Common source code” on page 233 lists source code of the REXX EXECs and shell scripts common, regardless of Linux distribution.

- Appendix C, “Source code specific to Linux” on page 249 lists source code of the REXX EXECs and shell scripts common, regardless of Linux distribution.

2

Planning

"*The only reason for time is so that everything doesn't happen at once.*"

— Albert Einstein

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

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

2.1 Bill of materials

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

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

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

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

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

  - Open Systems Adapter (OSA) network cards: One card minimum with 12 device numbers (technically 9, but OSA “triplets” usually start on an even address). Two OSA Express cards with eight device numbers on one and four on the other is recommended for high availability.
- A network-attached computer that will act as an Network File System (NFS) server temporarily with at least 6 GB of disk space, but more may be needed. A Linux PC or UNIX server is recommended.
- A workstation or desktop that has network access to the mainframe

2.1.2 Software resources

The following software resources are needed:

- z/VM 5.2 install media with documentation (DVD install is described in this book.)
- Linux install media (SLES or RHEL)
- An operating system for the NFS server
- The code associated with this book
- Tools on the workstation and desktop:
  - A 3270 Emulator such as *Attachmate Extra*, Hummingbird *Host Explorer*, or IBM *Personal Communications* for Windows desktops (for Linux desktops, a 3270 emulator named *x3270* is available)
  - A Linux SSH client such as PuTTY (recommended) or TeraTerm (for Linux desktops the `ssh` client is built in)
  - 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 should be routed to the OSA card(s).

2.2 z/VM conventions

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

2.2.1 Volume labeling convention

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

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

Sometimes DASD is shared among LPARs in which case your z/VM LPAR can *see* DASD *owned* by other LPARs. In this situation, it is convenient to identify the LPAR that *owns* the DASD. Therefore the volume labeling convention used in this book identifies the LPAR via the first character. That leaves the second character in the label to identify the basic function of the DASD.

The LPAR used in this book is identified by the character *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*.

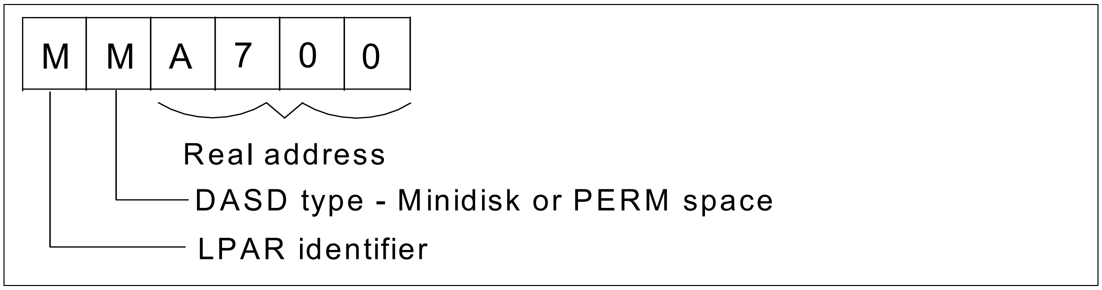


Figure 2-1 DASD labeling convention

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

2.2.2 Backup file naming convention

It is recommend that you keep copies of important z/VM and Linux configuration files. You should always keep copies of original configuration files in case you need to go back to them. Since z/VM file names are limited to 16 characters (eight for the file name and eight for the file type), only the last four characters of the file type are used. This often requires some characters to be overwritten. For the original file, the suffix `ORIG` is used, and for the most recent working copy, the suffix `WRKS` (for “it WoRKS”!) is used. For example, the original `USER DIRECT` file is copied to the 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’re looking for. The default Linux shell, **bash**, does this by default with the up arrow and down arrow keys.

There is a convention in z/VM to use the **F12** function key (labeled PF12 on physical 3270 devices) to retrieve the last command, though it is not defined to all user IDs. There is no convention retrieve commands in the other direction but it is possible to set another key to that function. Therefore, **F11** is used to *retrieve forward* since it is right next to F12. Also, the same function is useful in the editor, **XEDIT**. The **?** subcommand retrieves past commands, so it is recommended that you assign it to **F12**.

2.3 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 and does not discuss FBA disks accessed over SCSI/FCP. This is not because either technology is superior, but simply because DASD seems to be much more common than SCSI/FCP disks. If you were to use SCSI/FCP disks, cloning via 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 2.3GB, while 3390-9s are three times the size or about 6.9MB. Either size will work, though 3390-3s have been recommended over 3390-9s by some performance analysts. This book describes mainly using 3390-3s, however, comments are added where using 3390-9s differs - especially with installing z/VM.

Disk storage per Linux image

A single 3390-3, or 3339 cylinders, is allocated to the root file system (/) for each Linux image. Often recommendations are made to create many more file systems to be mounted over other directories, such as /usr/, /var/, /tmp/, /home/, /opt/, etc. This model is not used for two reasons:

- It is difficult to come up with a default set of file system sizes given that the Linux virtual servers will be running many different applications and workloads after being cloned.
- Under the philosophy of *keep things as simple as possible*.

One common argument for having many mounted file systems is so that the root file system will not fill up. To help prevent this from happening, it is recommended to add mounted file systems after cloning under the directory where data and applications will be added. For example, in the chapters that follow, a /nfs/ file system is added to the controller to store the install tree and a /home/ file system is added to LINUX02 to store user’s data.

2.4 Memory planning

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

Discussing memory planning further is outside the scope of this book (and the authors' knowledge :)). The following resources are recommended:

- The Redbook *Linux on IBM zSeries and S/390: Performance Measurement and Tuning*, 2003, on the Web at:

  `http://www.redbooks.ibm.com/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 simple rule that can be recommended is to have as few virtual machines logged on (or disconnected) as possible. Every virtual machine that can be logged off will mean more memory for the other virtual servers.

2.5 Password planning

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

This book considers different system administration roles:

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

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

The method of backing up z/VM data onto the Linux controller means that the Linux administrator will have access to all z/VM passwords. Therefore, the examples in this book set all z/VM and Linux system administration passwords to the same value, `lnx4vm`. If the z/VM and Linux system administrator roles must be kept separate and the Linux administrator is not to have access to the z/VM passwords, then a different method of backing up z/VM data must be chosen.

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

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

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

2.6 Planning worksheets

Four worksheets are included in this section. They are populated with the resources used in writing this book. There are also four corresponding blank worksheets in 2.7, “Blank worksheets” on page 17.

2.6.1 z/VM resources used in this book

Table 2-1 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 2-1 z/VM resources worksheet

Name	Value	Comment
LPAR name	P21	3 GB central storage/1 GB expanded, 4 shared IFLs
CPC name	PELCP01	Name of CPC on which the LPAR is located
z/VM system name	LNXVM52	Name to be assigned to z/VM system
TCP/IP host name	lat120	Assigned by a network administrator; helpful to set in DNS beforehand, but not necessary
TCP/IP domain name	pbm.ihost.com	Helpful to set in DNS beforehand
TCP/IP gateway	129.40.178.254	The router to and from the local subnet
DNS server 1	129.40.106.1	Assigned by the network administrator
DNS server 2/3 (optional)		Not used
OSA device name	eth0	Name of the interface to be assigned by IPWIZARD
OSA starting device number	3000	Start of OSA *triplet* for the z/VM TCP/IP stack
TCP/IP address	129.40.178.120	The TCP/IP address of the z/VM system
Subnet mask	255.255.255.0	Assigned by network administrator
OSA device type	QDIO	Often “QDIO” for OSA/Express cards
Network type	Ethernet	Usually “Ethernet”
Port name (optional)		Not required by z/VM
Router type	None	Usually “None”
Primary OSA device number for VSWITCH	3004	Specify the first device number (must be even number) and the next two device numbers will also be used
Secondary OSA device number for VSWITCH	3008	Should be on a different CHPID/OSA card

2.6.2 z/VM DASD used in this book

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

Table 2-2 z/VM DASD worksheet

Device number	Label	Type	Notes
A700	520RES	CP owned	z/VM System residence volume
A701	520SPL	System spool	Spool volume 1 from z/VM installation
A702	520PAG	CP owned	Paging volume 1 from z/VM installation
A703	520W01	System spool	z/VM W01 volume
A704	520W02	System paging	z/VM W02 volume
A705	MPA705	System paging	Paging volume 2
A706	MPA706	System paging	Paging volume 3
A707	MPA707	System paging	Paging Volume 4
A708	MPA708	System paging	Paging Volume 5
A709	MPA709	System paging	Paging volume 6
A728	MMA728	System minidisk	RHEL5 100 - the *master image*
A729	MMA729	System minidisk	RHEL 5 200 - the *controller*
A72A	MMA72A	System minidisk	RHEL 5 203 and LNXMAINT 191 and 192
A72B	MMA72B	System minidisk	RHEL 5 204, part of the `/nfs/` logical volume
A72C	MMA72C	System minidisk	RHEL 5 205, part of the `/nfs/` logical volume
A72D	MMA72D	System minidisk	RHEL 5 206, part of the `/nfs/` logical volume
A72E	MMA72E	System minidisk	RHEL 5 207, part of the `/nfs/` logical volume
A740	MMA740	System minidisk	LINUX01 100
A741	MMA741	System minidisk	LINUX02 100
A742	MMA742	System minidisk	LINUX03 100
A743	MMA743	System minidisk	LINUX04 100
A744	MMA744	System minidisk	LINUX05 100
A745	MMA745	System minidisk	LINUX06 100
A746	MMA746	System minidisk	LINUX07 100

2.6.3 Linux resources used in this book

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

Table 2-3 Linux resources worksheet

Name	Value	Comment
Linux install password	lnx4vm	

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

2.6.4 Linux user IDs used in this book

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

Table 2-4 Linux user ID worksheet

Linux user ID	IP address	DNS name	Notes
RHEL5	129.40.178.128	lat128.pbm.ihost.com	A 64-bit controller (cloner) and master (golden) image
LINUX01	129.40.178.121	lat121.pbm.ihost.com	A Web virtual server
LINUX02	129.40.178.122	lat122.pbm.ihost.com	An LDAP virtual server
LINUX03	129.40.178.123	lat123.pbm.ihost.com	A file and print virtual server
LINUX04	129.40.178.124	lat124.pbm.ihost.com	An application development server
LINUX05	129.40.178.125	lat125.pbm.ihost.com	Available for use
LINUX06	129.40.178.126	lat126.pbm.ihost.com	Available for use
LINUX07	129.40.178.127	lat127.pbm.ihost.com	A test server for kickstart

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 2-5 to document the z/VM resources that you will use.

Table 2-5 z/VM resources blank worksheet

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

2.7.2 z/VM DASD worksheet

Use the worksheet in Table 2-6 to document the z/VM DASD that you will use.

Table 2-6 z/VM DASD blank worksheet

Device number	Label	Type	Notes

2.7.3 Linux resources worksheet

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

Table 2-7 Linux resources blank worksheet

Name	Value	Comment
NFS server TCP/IP address		
Linux install password		
Linux 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 2-8 to document the Linux user IDs that you will create.

Table 2-8 Linux user ID blank worksheet

Linux user ID	IP address	DNS name	Notes

3

Configuring a desktop machine

"*Technological progress is like an axe in the hands of a pathological criminal.*"

— Albert Einstein

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.

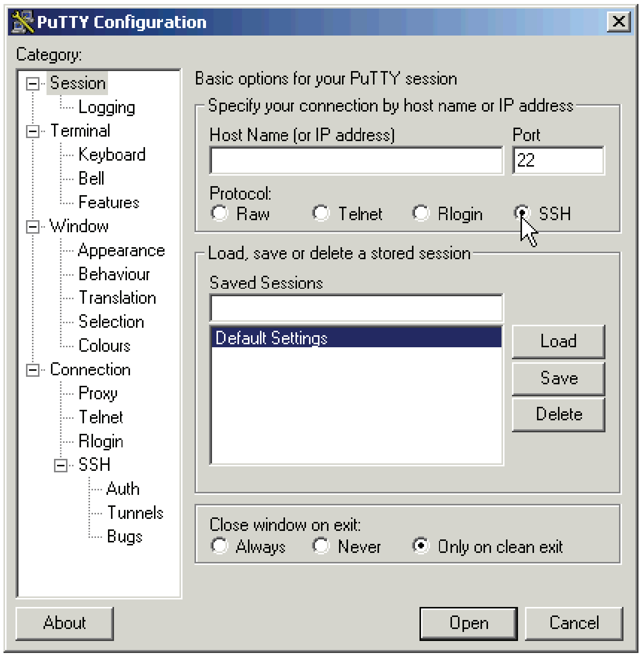
PuTTY is probably the most commonly used. You can find a PuTTY client for Windows on CD1 of a SLES 9 or 10 distribution in the `/dosutils/putty` directory. You can download PuTTY from the Web at:

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

To download from this page, click on the **putty.exe** link for your architecture. Save the file in a directory path such as `C:\WINNT`. PuTTY is a stand-alone 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 3-3 should open. If you spend a few minutes to configure PuTTY it may pay off in time savings.

1. In the *PuTTY Configuration* window, in the left Category panel, click **SSH**.
2. Under the Protocol heading on the right, click the **SSH** radio button. This specifies to use the SSH protocol.



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address)

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

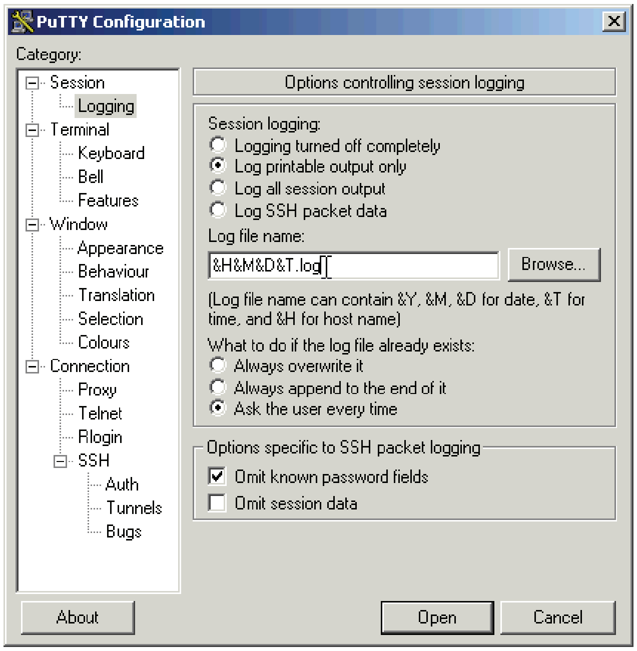
Load, save or delete a stored session

Saved Sessions: Default Settings

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

Figure 3-1 PuTTY Configuration window

3. Click on **`Logging`** in th left panel. Click the radio button **Log printable output only**. This will allow you to go back and check on the output of certain commands. Set the *Log file name* to **&H&M&D&T.log** so it will include a timestamp in the file name.



Options controlling session logging

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

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

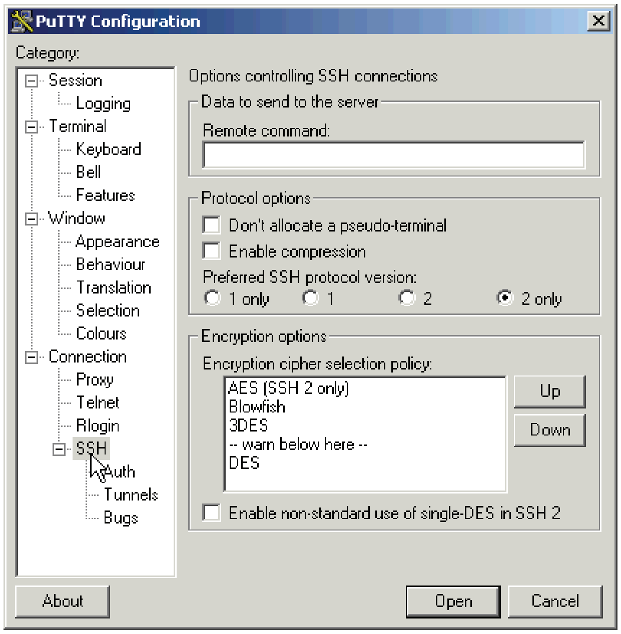
(Log file name can contain &Y, &M, &D for date, &T for time, and &H for host name)

What to do if the log file already exists:
○ Always overwrite it
○ Always append to the end of it
● Ask the user every time

Options specific to SSH packet logging
☑ Omit known password fields
□ Omit session data

Figure 3-2 Setting Logging

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



Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

□ Don't allocate a pseudo-terminal

□ Enable compression

Preferred SSH protocol version:

○ 1 only ○ 1 ○ 2 ● 2 only

Encryption options

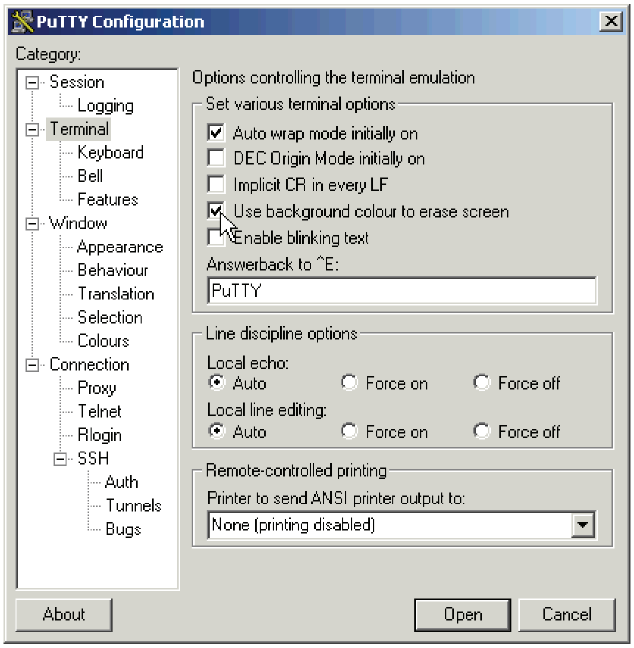
Encryption cipher selection policy:

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

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

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

6. In the left Category panel, click **Terminal**.
7. Select the **Use background colour to erase screen** check box, which results in a better job of painting the screen for applications that uses curses (block graphics).



Options controlling the terminal emulation

Set various terminal options

☑ Auto wrap mode initially on

□ DEC Origin Mode initially on

□ Implicit CR in every LF

☑ Use background colour to erase screen

□ Enable blinking text

Answerback to ^E: PuTTY

Line discipline options

Local echo: ● Auto ○ Force on ○ Force off

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

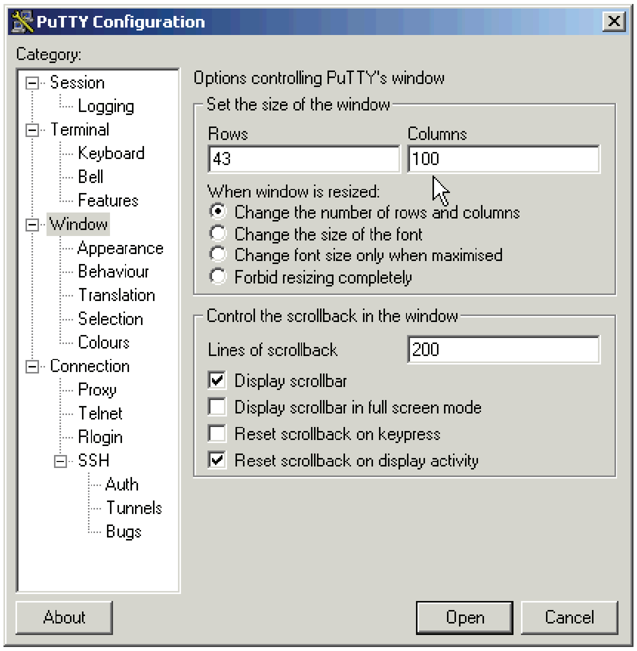
Remote-controlled printing

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

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

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

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



Options controlling PuTTY's window

Set the size of the window

Rows: 43

Columns: 100

When window is resized:

● Change the number of rows and columns
○ Change the size of the font
○ Change font size only when maximised
○ Forbid resizing completely

Control the scrollback in the window

Lines of scrollback: 200

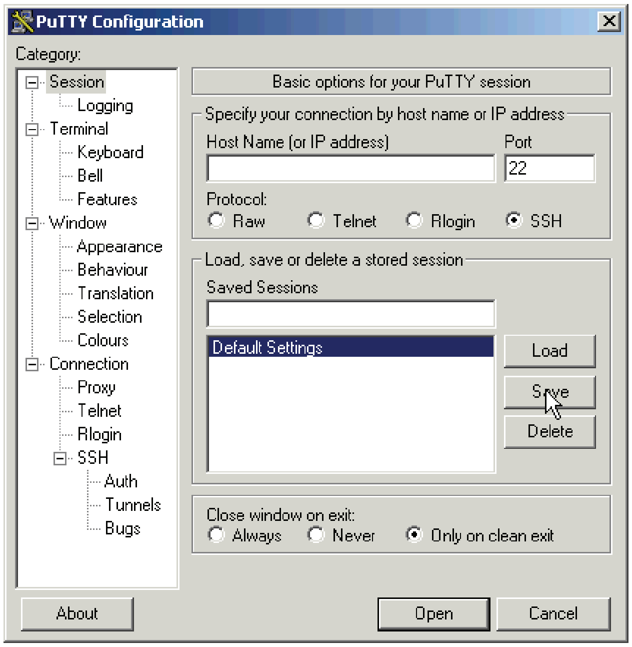
☑ Display scrollbar
☐ Display scrollbar in full screen mode
☐ Reset scrollback on keypress
☑ Reset scrollback on display activity

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

Saving sessions

1. Click **Session** to get back to the initial window when PuTTY is invoked (see Figure 3-6).

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



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address):

Port: 22

Protocol:

○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:

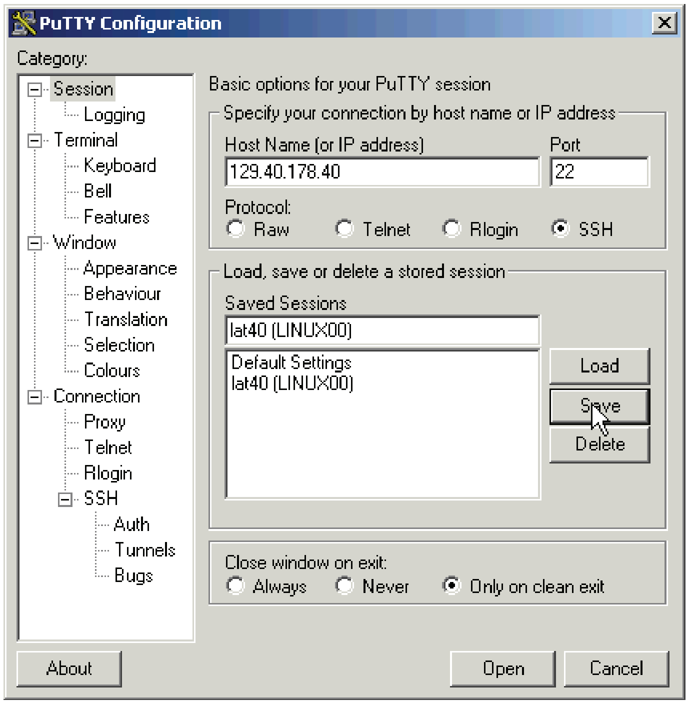
○ Always ○ Never ● Only on clean exit

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

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

1. In the Host Name (or IP address) field, enter the TCP/IP address (or DNS name).
2. Under Saved Sessions, choose a name that you will remember. In this example, the name `lat40 (LINUX00)` is used. This is the DNS name and the z/VM user ID.
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 is invoked. The final window should appear similar to that shown in Figure 3-7.



Host Name (or IP address): 129.40.178.40

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions: lat40 (LINUX00)

Default Settings
lat40 (LINUX00)

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

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

3.2 Setting up a VNC client

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

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

3.2.1 Downloading and running RealVNC

If you have a Windows desktop, the VNC client from RealVNC is a popular choice (you can also find a TightVNC client for Windows on CD1 of a SLES 9 or 10 distribution in the `/dosutils/tightvnc` directory).

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

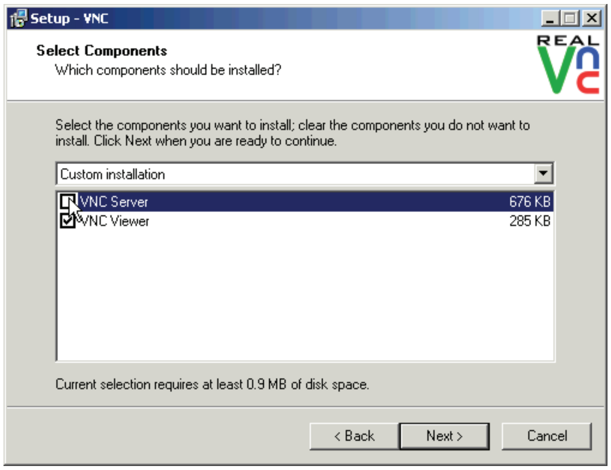
`http://www.realvnc.com`

The download page is:

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

Fill out the Web form and download the executable. When you have downloaded it, run it and an install program will start. At the time of writing of this book, RealVNC 4.1.1 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.



Select Components

Which components should be installed?

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

Custom installation

□ VNC Server 676 KB

☑ VNC Viewer 285 KB

Current selection requires at least 0.9 MB of disk space.

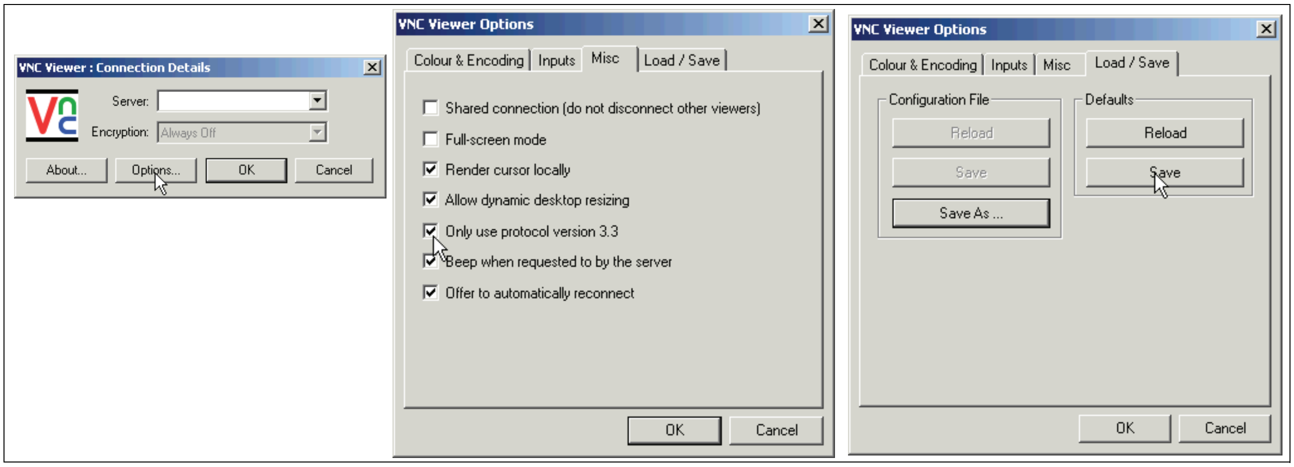
Figure 3-8 RealVNC Select Components panel

Complete the screens and the installation process should go quickly.

3.2.2 Customizing RealVNC

The latest VNC protocol is version 4, which is the default with the VNC client. This version will work with the VNC servers shipped with SLES 10 or RHEL5, however, version 3.3 is required to communicate with the VNC server shipped with SLES 9.

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



Server:

Encryption: Always Off

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

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

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

3.3 3270 emulators

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

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

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

- Set the **Enter** and **Clear** function keys to be where you would expect them. On some emulators, the default Enter key action is set to the right `Ctrl` key of modern keyboards. Likewise the Clear key action is sometimes set to the `Esc` key in the upper left corner of modern keyboards or the **Pause** key in the upper right.
- Set a larger screen. Often the default number of lines in an emulator session is 24. You will probably be more productive with a 32 or 43 lines if that can easily fit in a window given your desktop display size and resolution.
- Have the session automatically reconnect after logoff. Having a new logon screen 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

"*Example isn't another way to teach. It is the only way to teach.*"

— Albert Einstein

To complete this chapter, you must complete the majority of chapter 6 *Configuring an NFS server*. However, it is recommended that you start here, because there is a step when installing z/VM (**instdvd**) that takes over two hours or possibly much longer near the end of section 4.1.2, "Copying a vanilla z/VM system to DASD" on page 32. While that process is running, you can configure the Network File System (NFS) server. Alternatively, if you have other personnel who can work on the project, you can start both chapters at the same time on the different systems.

4.1 Installing z/VM from DVD

The section that follows assumes a first level installation of z/VM from DVD onto DASD. If you have not already done so, complete the worksheet in 2.7.1, "z/VM resources worksheet" on page 17. You will need access to the Hardware Management Console (HMC) with a user ID that has authority to go into single object operations mode.

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

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

If you are not familiar with the HMC and z/ VM, you may want to use the complete installation manual *z/VM Guide for Automated Installation and Service Version 5 Release 2.0*, GC24-6099. If you are installing z/VM at the *second level* (z/VM under z/VM) or onto SCSI disk, you will want to use the z/VM manaul as the sections that follow don't address these options.

4.1.1 Booting z/VM from DVD

This section explains how to install z/VM 5.2 from an HMC with a DVD-ROM onto 3390-3 DASD. Some words are included for installing onto 3390-9s. For alternative configurations such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

1. On the Hardware Management Console, select the LPAR on which you want to install z/VM.
2. If necessary, click the *racetrack* buttons (two buttons that are circular arrows on the bottom right corner) to traverse to the *CPC Recovery* (sometimes named just *Recovery*) menu.
3. On the *CPC Recovery* menu, double-click the **Integrated 3270 Console** as shown at the bottom of Figure 4-1. A window entitled *Integrated 3270 Console for <yourCPC>* 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.

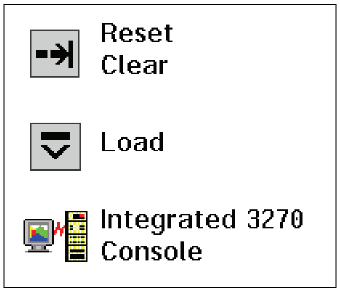


Figure 4-1 Integrated 3270 Console icon

4. Place the **z/VM Product Package Verson 5 Release 2.0** DVD in the HMC DVD drive.
5. Get into *Single Object Operations* mode. To get into this mode, perform the following steps:
   a. Double-click the **Groups** icon in the *Views Area*
   b. Double-click **Defined CPCs** in the *Groups Work Area*.
   c. Select your CPC.
   d. If necessary, go around the *racetrack* (the buttons with circular arrows on the bottom right corner) to the *CPC Recovery* menu.
   e. Double-click the **Single Object Operations** icon. Click **yes** to confirm. 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. If so, click **OK.**
   f. Double-click **Groups** near the top of this window.
   g. Double-click **Images** in the Groups Work Area.

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

6. Select the LPAR that z/VM will be installed into.
7. Go around the racetrack in this window to the CPC Recovery menu. Double-click the **Load from CD-ROM or Server** icon when you see it (see Figure 4-2 on page 31).

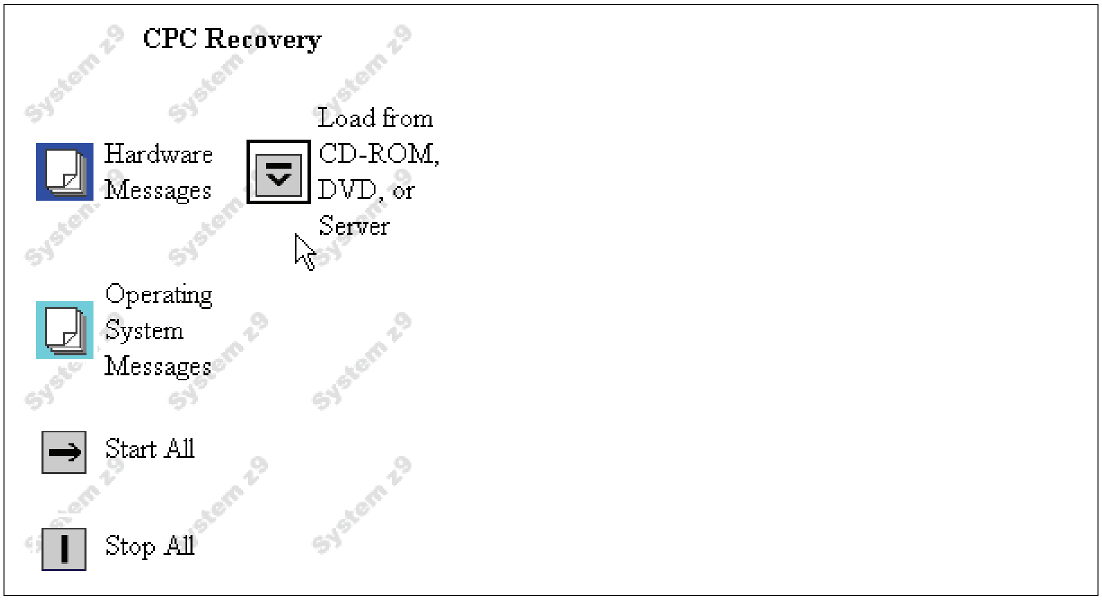
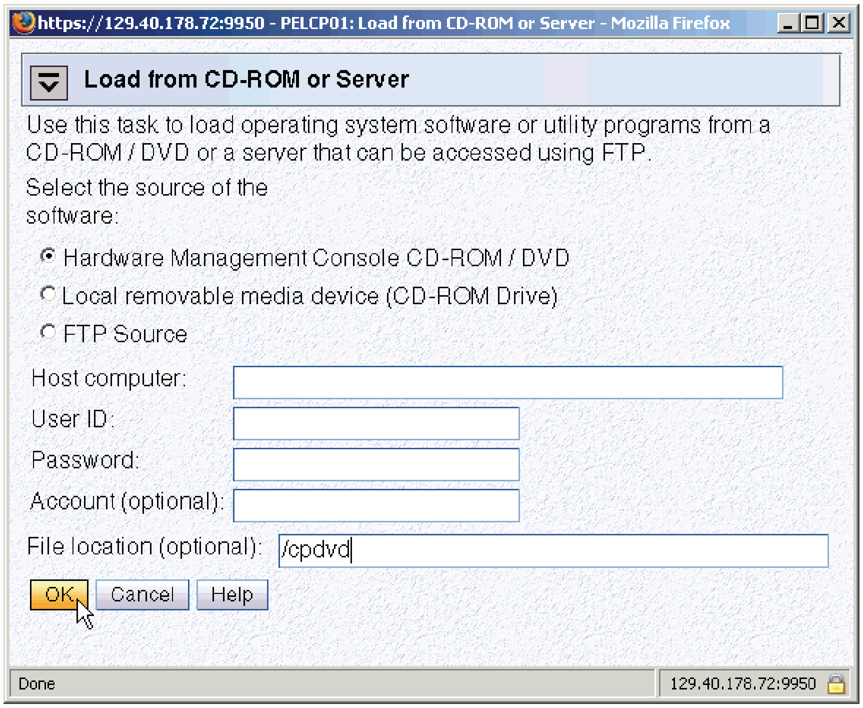


Figure 4-2 CPC Recovery menu with "Load from CD-ROM or Server" icon present.

8. On the *Load CD-ROM or Serve*r window as shown in Figure 4-3, the radio button **Hardware Management Console CD-ROM/DVD** should be selected.
9. In the same *Load CD-ROM or Serve*r window, fill in *File Location* with **/cpdvd**. This is the directory on the DVD with the z/VM 5.2 installation code.
10. Click **OK**.


https://129.40.178.72:9950 - PELCP01: Load from CD-ROM or Server - Mozilla Firefox

Load from CD-ROM or Server

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

Select the source of the software:

● Hardware Management Console CD-ROM / DVD
○ Local removable media device (CD-ROM Drive)
○ FTP Source

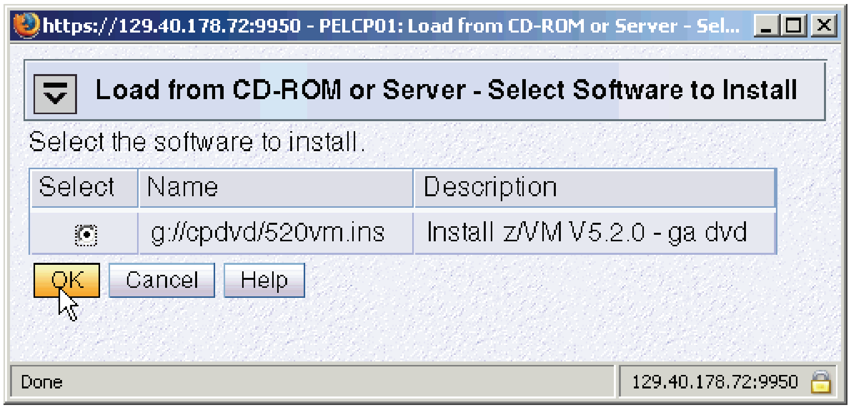
Host computer:
User ID:
Password:
Account (optional):
File location (optional): /cpdvd

OK
Cancel
Help
Done
129.40.178.72:9950

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

11. Load the RAMDISK:

a. From the *Load from CD-ROM or Server* panel, the software **520vm.ins** should be selected as shown in Figure 4-4. Click **OK**.



https://129.40.178.72:9950 - PELCP01: Load from CD-ROM or Server - Sel...

Load from CD-ROM or Server - Select Software to Install

Select the software to install.

Select	Name	Description
●	g://cpdvd/520vm.ins	Install z/VM V5.2.0 - ga dvd

Figure 4-4 Selecting z/VM 5.2 RAMdisk system

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

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

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

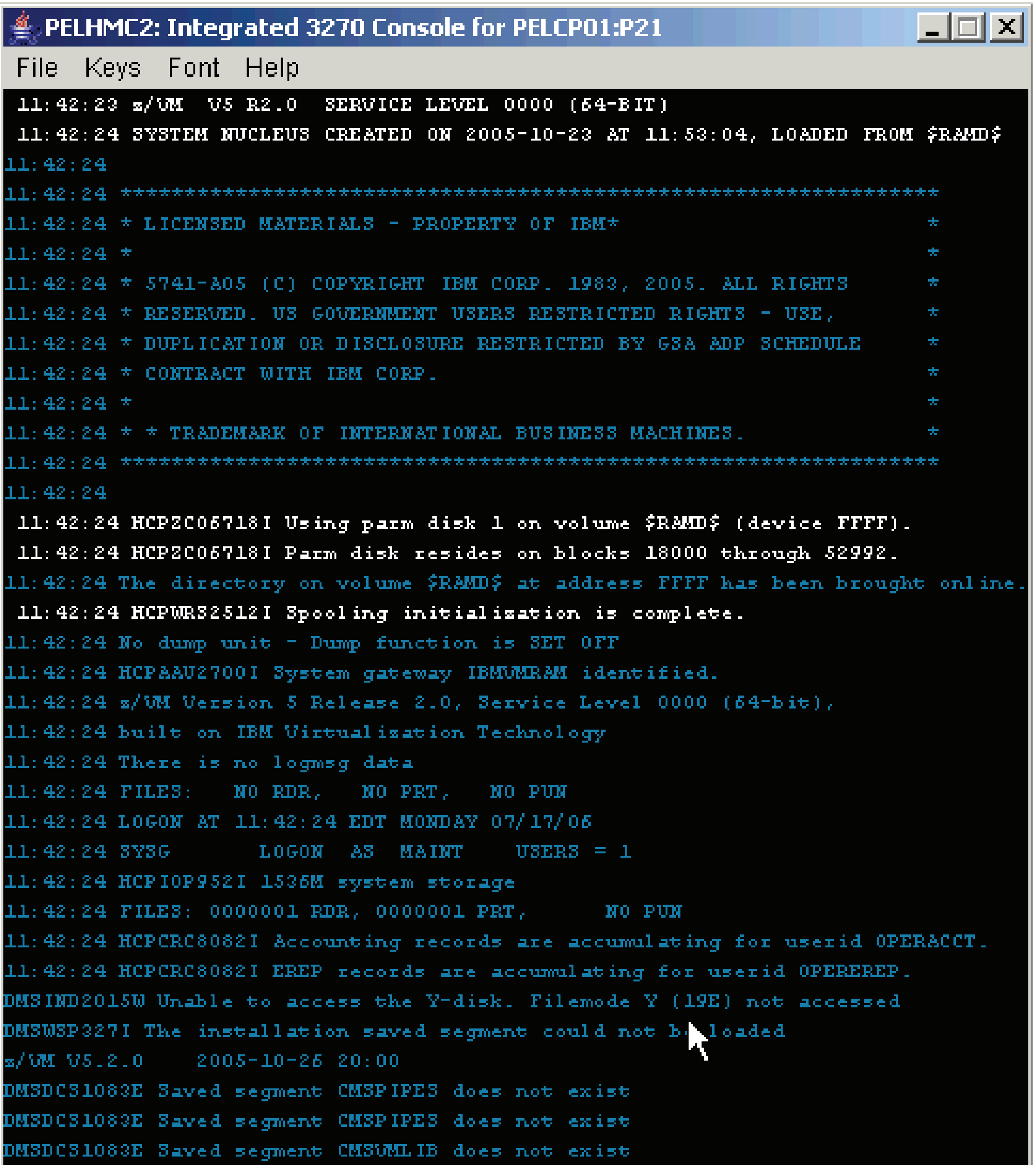
You should now have an in-memory z/VM 5.2 system running.

4.1.2 Copying a vanilla z/VM system to DASD

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

1. You can now get out of Single object operations mode. To do so, log off the primary SE window by closing that window.

2. Use the **Alt-Tab** sequence, move to the Integrated 3270 Console window. The RAMdisk IPLs and the system comes up with the `MAINT` user ID logged on. You should see z/VM boot as shown in Figure 4-5:

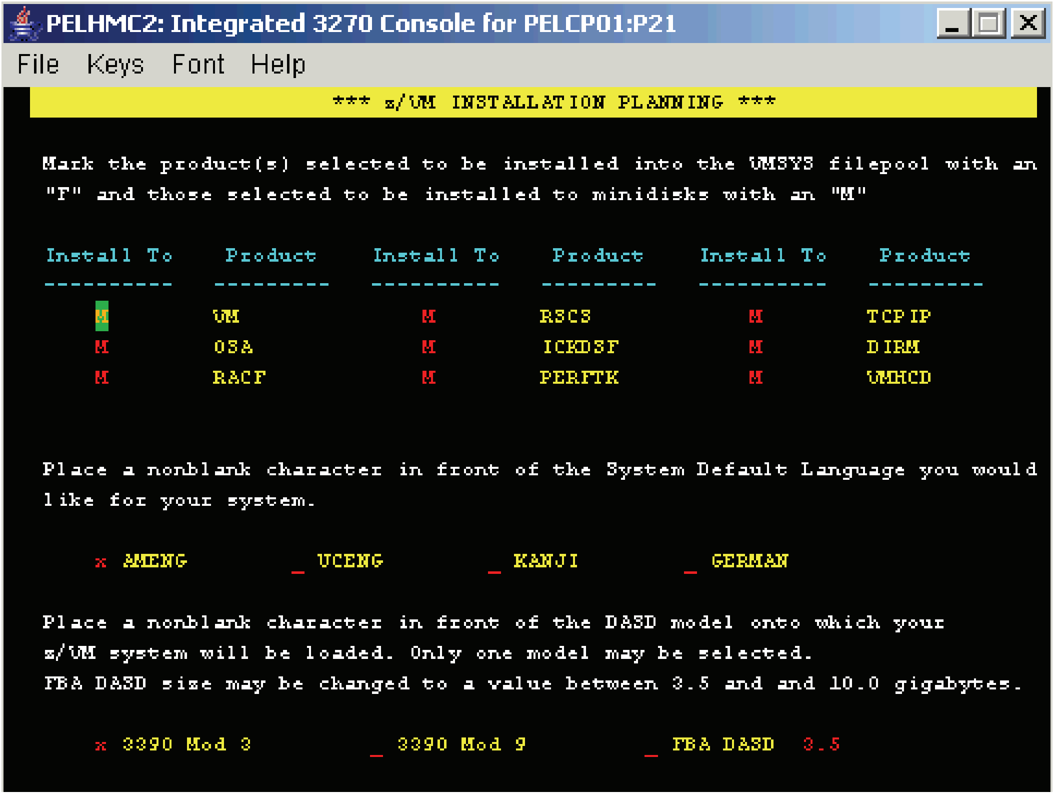


```
 11:42:23 z/VM  V5 R2.0  SERVICE LEVEL 0000 (64-BIT)
 11:42:24 SYSTEM NUCLEUS CREATED ON 2005-10-23 AT 11:53:04, LOADED FROM $RAMD$
11:42:24
11:42:24 ****************************************************************
11:42:24 * LICENSED MATERIALS - PROPERTY OF IBM*                         *
11:42:24 *                                                               *
11:42:24 * 5741-A05 (C) COPYRIGHT IBM CORP. 1983, 2005. ALL RIGHTS       *
11:42:24 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE,        *
11:42:24 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE      *
11:42:24 * CONTRACT WITH IBM CORP.                                       *
11:42:24 *                                                               *
11:42:24 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES.               *
11:42:24 ****************************************************************
11:42:24
 11:42:24 HCPZCO6718I Using parm disk 1 on volume $RAMD$ (device FFFF).
 11:42:24 HCPZCO6718I Parm disk resides on blocks 18000 through 52992.
11:42:24 The directory on volume $RAMD$ at address FFFF has been brought online.
 11:42:24 HCPWRS2512I Spooling initialization is complete.
11:42:24 No dump unit - Dump function is SET OFF
11:42:24 HCPAAU2700I System gateway IBMVMRAM identified.
11:42:24 z/VM Version 5 Release 2.0, Service Level 0000 (64-bit),
11:42:24 built on IBM Virtualization Technology
11:42:24 There is no logmsg data
11:42:24 FILES:   NO RDR,   NO PRT,   NO PUN
11:42:24 LOGON AT 11:42:24 EDT MONDAY 07/17/06
11:42:24 SYSG       LOGON  AS  MAINT    USERS = 1
11:42:24 HCPIOP952I 1536M system storage
11:42:24 FILES: 0000001 RDR, 0000001 PRT,      NO PUN
11:42:24 HCPCRC8082I Accounting records are accumulating for userid OPERACCT.
11:42:24 HCPCRC8082I EREP records are accumulating for userid OPEREREP.
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
DMSWSP327I The installation saved segment could not be loaded
z/VM V5.2.0    2005-10-26 20:00
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSPIPES does not exist
DMSDCS1083E Saved segment CMSVMLIB does not exist
```

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

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

```
==> instplan
```



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

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

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


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

    x AMENG          _ UCENG          _ KANJI          _ GERMAN

Place a nonblank character in front of the DASD model onto which your
z/VM system will be loaded. Only one model may be selected.
FBA DASD size may be changed to a value between 3.5 and and 10.0 gigabytes.

    x 3390 Mod 3            _ 3390 Mod 9            _ FBA DASD  3.5
```

Figure 4-6   Installation planning panel

4. You should see the display as shown in Figure 4-6. It is recommended that you leave the M's in the top section alone.
5. 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 above.
6. Press **F5**. You should the message `HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY`.
7. 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 18. In this example, the devices are `a700-a704`.

   ```
   ==> att <a700-a704> *
   a700-a704 ATTACHED TO MAINT
   ```

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

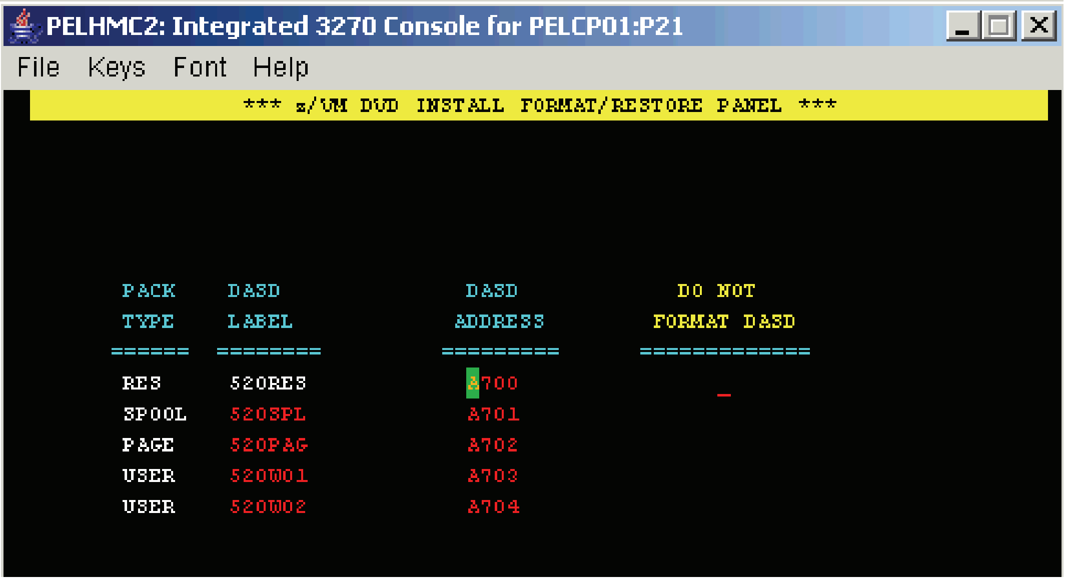
Running INSTDVD

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

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

```
==> instdvd
```

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



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

   PACK     DASD               DASD           DO NOT
   TYPE     LABEL            ADDRESS        FORMAT DASD
  ======  ========          =========      =============
   RES      520RES            A700               _
   SPOOL    520SPL            A701
   PAGE     520PAG            A702
   USER     520W01            A703
   USER     520W02            A704
```

Figure 4-7 INSTDVD DASD address panel

   a. Enter the addresses of the five volumes that z/VM will be installed on.

   b. Do *not* select the DO NOT FORMAT DASD check box on the right side of the panel.

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

3. When you see the question `DO YOU WANT TO CONTINUE?`, type **`Y`.** You should see the message `NOW FORMATTING DASD <A700>`.

> **Important:** INSTDVD takes about two and one half 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.

4. You are asked to place the system RSU in the drive. Insert the **z/VM Stacked Recommended Service Upgrade 5201** DVD into the HMC DVD-ROM drive and type **`GO`**. You should see a messages of the form `DVDLOAD: LOADING FILE CKD5000x IMAGE *`. This step takes two to four minutes.

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

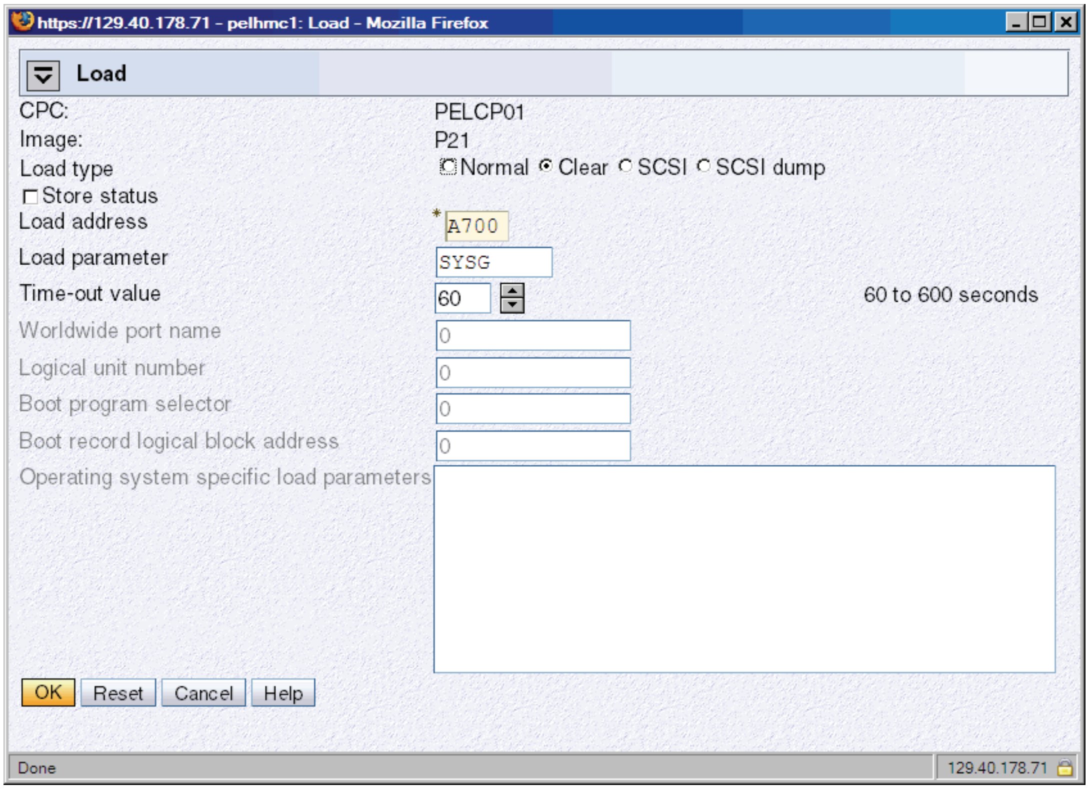
4.1.3 IPL the vanilla z/VM from DASD

IPL your initial z/VM system now on DASD.

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

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

3. The Load window opens as shown in Figure 4-8. Follow these steps:
   a. Check the radio button **Clear**.
   b. Set the load address to the new system residence (520RES) volume which is **<A700>** in this example.
   c. Set the load parameter to **SYSG**. This specifies to use the Integrated 3270 console.
   d. Click **OK** to IPL.



https://129.40.178.71 - pelhmc1: Load - Mozilla Firefox

Load

CPC: PELCP01

Image: P21

Load type ○ Normal ● Clear ○ SCSI ○ SCSI dump

□ Store status

Load address * A700

Load parameter SYSG

Time-out value 60 60 to 600 seconds

Worldwide port name 0

Logical unit number 0

Boot program selector 0

Boot record logical block address 0

Operating system specific load parameters

Figure 4-8 Load window

4. When you see the *Load Task Confirmation* window, click **Yes**.
5. After 1-3 minutes you should see *Success* in the *Load Program* window. Click **OK**.
6. Use the **Alt-Tab** sequence to move back to the Integrated 3270 console window. You should see the *Standalone Program Loader* panel as shown in the following diagram.
   a. Press the **Tab** key to traverse to the *IPL Parameters* section and enter the value **`cons=sysg`**. This specifies to use the Integrated 3270 console.

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

 DEVICE NUMBER:   A700       MINIDISK OFFSET:   00000000   EXTENT:  1

 MODULE NAME:     CPLOAD     LOAD ORIGIN:       2000

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


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




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



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

b. Press the **F10** key to continue the IPL of your z/VM system. This should take around 2-3 minutes.

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

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

8. At the Change TOD clock prompt enter:

```
==> no
```

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

```
==> disc
```

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

4.1.4 Completing the z/VM installation

Follow these steps to complete the z/VM installation

1. On the z/VM login screen, logon as **`MAINT.`** The password is **`MAINT`**. You may receive messages `HCPLMN108E` or `DMSACP113S` about disks not linked or attached. This is not a problem. Press **Enter** when you see the `VM Read` prompt in the lower right corner.

> **Important:** When logging onto a z/VM user ID that runs CMS, you should usually press **Enter** at the `VM READ` prompt. This will result in a prompt of the form:
>
> ```
> Ready; T=0.01/0.01 11:14:20
> ```

2. IPL CMS and press **Enter**. You should see the `Ready;` prompt.

```
==> ipl cms
```

```
==> Press Enter
Ready;
```

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

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

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

4. Load the recommended service. For z/VM 5.2, the service name is **5201RSU1**. Run the following commands:

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

   This step should take about 3-6 minutes. The last message should be `VMFSRV2760I SERVICE processing completed successfully`.

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

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

   This step should take about 2-4 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.

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

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

7. You will lose your 3270 session. 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 screen.

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 via the **IPWIZARD** command which is generally used just once. After **IPWIZARD** creates the initial configuration files, they are typically maintained manually.

From the z/VM logon 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 screen.

```
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

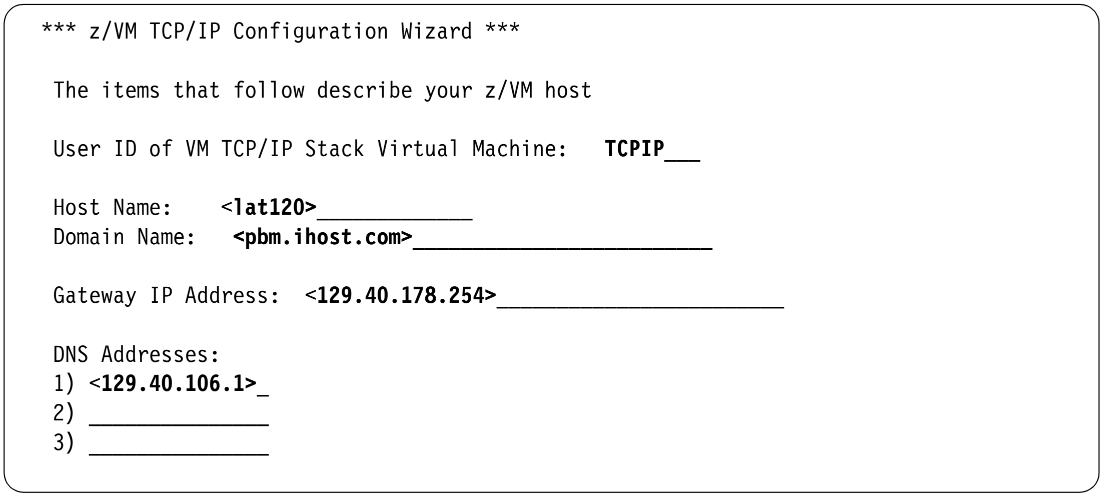
The **`IPWIZARD`** command is on the `MAINT 193` disk. You will need to access it file mode G via the **`ACCESS`** command so you will pick up **`IPWIZARD`** from that minidisk.

1. Access the `MAINT 193` disk:

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

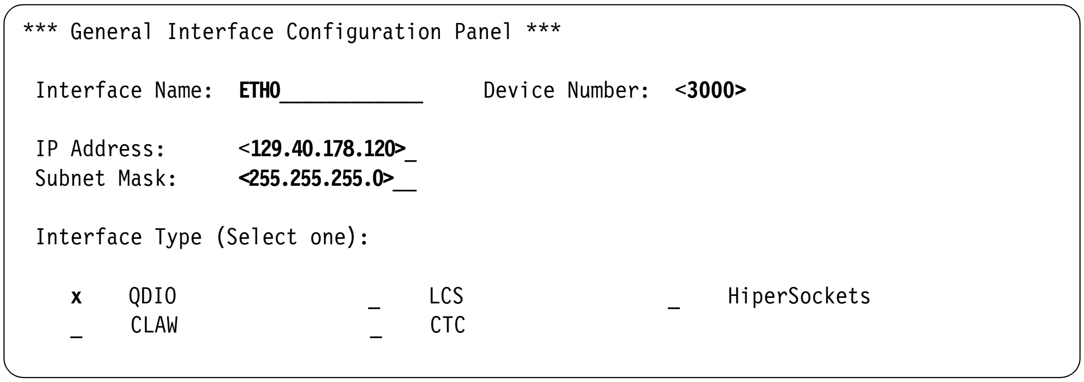
2. Invoke **`IPWIZARD`**.

   ```
   ==> ipwizard
   ```



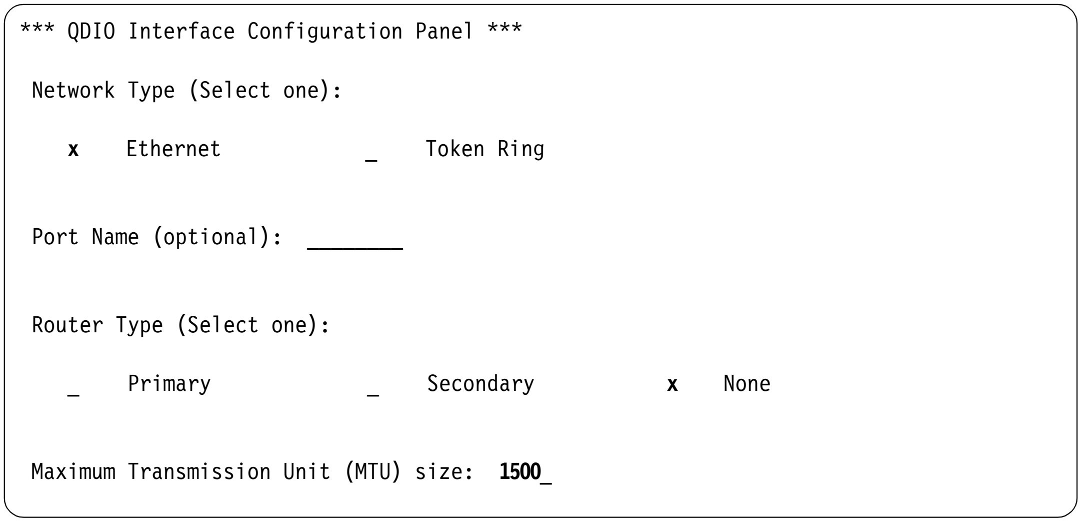
```
*** z/VM TCP/IP Configuration Wizard ***

 The items that follow describe your z/VM host

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

 Host Name:    <lat120>____________
 Domain Name:   <pbm.ihost.com>________________________

 Gateway IP Address:  <129.40.178.254>_______________________

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

3. The *z/VM TCP/IP Configuration Wizard* opens as shown in the preceding example. The first field, User ID, should always be **TCPIP**. Obtain the remaining values from the 2.7.1, “z/VM resources worksheet” on page 17 and press **F8**.



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

 Interface Name:  ETH0____________     Device Number:  <3000>

 IP Address:      <129.40.178.120>_
 Subnet Mask:     <255.255.255.0>__

 Interface Type (Select one):

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

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



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

 Network Type (Select one):

    x    Ethernet            _    Token Ring


 Port Name (optional):  ________


 Router Type (Select one):

    _    Primary             _    Secondary           x    None


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

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

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

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

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 2     FORCED BY MAINT
...
Successfully PINGed Interface (129.40.178.120)
Successfully PINGed Gateway (129.40.178.254)
Ping Level 520: Pinging host 129.40.106.1.
                Enter 'HX' followed by 'BEGIN' to interrupt.
```

Important: If the DNS server cannot be pinged, enter **1** to try it again.

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

Watch for the message **`IPWIZARD EXEC ENDED SUCCESSFULLY`**.

```
...
Successfully PINGed Interface (129.40.178.120)
Successfully PINGed Gateway (129.40.178.254)
Successfully PINGed DNS (129.40.106.1)
DTCIPW2519I Configuration complete; connectivity has been verified
```

```
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCPIP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

7. 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** 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 via a 3270 emulator. See 3.2, “3270 emulators” on page 17 for some brief words on that subject.

If you want to switch to 3270 emulator, you can **LOGOFF** of MAINT or **DISConnect**. If you logoff 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 of of them will terminate the service or crash Linux.

4.3 Configuring the XEDIT profile

Logon to MAINT if you are not already.

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

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

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

Before:

```
SET PF12 FILE
```

After:

```
SET PF12 ?
```

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

4.4 Customizing the SYSTEM CONFIG file

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

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

To make these changes, perform the following steps:

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

```
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  MVA740 A740 CKD          39        158
MNTCF2 MAINT    0CF2  B   R/O  MVA740 A740 CKD         159        278
MNTCF3 MAINT    0CF3  C   R/O  MVA740 A740 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    0CF2  B   R/O  MVA740 A740 CKD         159        278
MNTCF3 MAINT    0CF3  C   R/O  MVA740 A740 CKD         279        398
```

2. Once it is released you are able to access the `MAINT CF1` disk read-write. Use the **LINK** command with multi-read (**MR**) parameter and **ACCESS** command to get read-write access to the minidisk.

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

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

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

4. The system name is set to `ZVMV5R20` by default in the `System_Identifier_Default` statement. You can search for it using the **/ XEDIT** subcommand:

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

Modify this to the new name of your system:

```
System_Identifier_Default <LNXVM52>
```

5. 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
```

6. Define a VSWITCH:

   Use the **BOTTOM** subcommand to go to the bottom of the file. Add some lines (you can use the **XEDIT** add subcommand **a3**). Define a VSWITCH and set the MAC address prefix. If you have multiple z/VM systems, each should have a unique prefix. Modify the two starting addresses of the OSA triplets (3004 and 3008 in this example) to those you specified in 2.7.1, "z/VM resources worksheet" on page 17.

```
====> bot
====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev <3004> <3008>
vmlan macprefix 020001
```

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

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

8. 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.

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

```
==> rel f (det
DASD 0CF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  MVA740 A740 CKD          39        158
MNTCF2 MAINT    0CF2  B   R/O  MVA740 A740 CKD         159        278
MNTCF3 MAINT    0CF3  C   R/O  MVA740 A740 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.

1. Logoff of MAINT.

   ```
   ==> log
   ```

2. You should see a new logon panel. **Logon to AUTOLOG1**. Again the password is the same as the user ID.

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

   ```
   LOGON AUTOLOG1
   z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES:   NO RDR,   NO PRT,   NO PUN
   LOGON AT 13:30:12 EST THURSDAY 01/19/06
   DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
   z/VM V5.2.0    2005-12-22 09:36
   acc (noprof
   ```

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

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

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

   b. Copy the PROFILE XEDIT to your A disk:

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

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

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

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

   a. The z/VM Shared File System (SFS), is not required to run Linux so you can safely delete the three lines that **XAUTOLOG** the user IDs VMSERVS, VMSERVR and VMSERVU.

   b. You can also safely delete the Address Command line.

   c. Add a line to start the TCPIP user ID via the **XAUTOLOG** command and keep two statements that start the VSWITCH controllers.

   d. Add a line to **logoff** of AUTOLOG1 when the EXEC is complete. There is no need to keep that virtual machine running as 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 DTCVSW1'             /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'             /* start VSWITCH controller 2 */
'cp logoff'                       /* logoff when done */
```

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

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

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

4.5.1 Renaming the TCPIP configuration file

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

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

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

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

You have now renamed your TCP/IP profile.

4.5.2 Copy the PROFILE XEDIT file

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

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

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

b. Copy the PROFILE XEDIT to your A disk:

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

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

4.5.3 Configuring the FTP server

It is recommend that you turn on the FTP server. To do so, edit the newly renamed configuration file and add an AUTOLOG statement near the top of the file with FTPSERVE as the only entry. In the PORT statement, remove the semicolons to uncomment the lines with FTPSERVE on them (ports 20 and 21). These changes will cause the FTP server to start when TCPIP is started. The important lines before the file is edited and after are shown:

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

Before:

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

After:

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

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

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

4.5.4 Shutting down and reIPLing the system

It is now time to shutdown and reIPL the system. You will also want to be able to shutdown and reIPL z/VM without having to access the HMC. Often, the HMC will be logged off and thus the Integrated 3270 console (SYSG) will not be available. Because of these factors it is useful to use the *System Console* (SYSC - which has a title of *Operating System Messages* on the HMC) in order to shut down z/VM and reIPL it without needing to use the console. This console is always accessible whether you are logged on to the HMC or not. z/VM messages during both the shutdown and reIPL process will be written to the system console, but often you will be able to ignore them - you just want your system back in a few minutes over the network.

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. You may want to watch the system console as z/VM shuts down and comes back up. When your system is back up, perform the following commands

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

2. Query the new VSWITCH:

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1  VDEV:  3004
  Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2  VDEV:  3008 BACKUP
```

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

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

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

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

4.6 Adding paging volumes

The z/VM operating system resides on the first three CP volumes (or one volume if installing onto 3390-9s). z/VM 5.2 now also installs with one full paging volume and one full spool volume. A single spool volume is probably adequate for Linux needs, however, a single paging volume is probably not.

It is recommended that you add five paging volumes so you will have a total of six (or one more 3390-9 if installing onto 3390-9s which will give the same total of 20034 cylinders). If you do not have sufficient DASD, this number can be reduced. Having adequate paging space will give you plenty of *headroom* to add more Linux virtual machines. A rule of thumb for the amount of paging space is to have twice as much as the total of all memory for all running Linux user IDs combined.

4.6.1 Formatting the paging volumes

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

Therefore, a REXX EXEC named **`CPFORMAT`** has been provided to allow you to format many volumes with a single command. The source code for this EXEC is in the section B.1.1, “The CPFORMAT EXEC” on page 233. 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 11. 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.

> **Important:** At this point, you will 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, do it now.

Getting the CPFORMAT EXEC to z/VM

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

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. Note that the directory name will be one of the following two depending on the Linux distribution you are working with:

```
# cd /nfs/virt-cookbook-S10/vm    // if you are working with SLES 10
# cd /nfs/virt-cookbook-RH5/vm    // if you are working with RHEL 5
# ls CPFORMAT*
CPFORMAT.EXEC
```

Now start an FTP session to z/VM. If you get a reply from the FTP server it shows that you correctly configured it on the z/VM `TCPMAINT` user ID.

```
# ftp <129.40.178.120>
220-FTPSERVE IBM VM Level 520 at LAT120.PBM.IHOST.COM, 14:53:44 EST WEDNESDAY 2004-12-08
Name (129.40.178.120:root): maint
Password: maint
...
ftp> put CPFORMAT.EXEC
...
ftp> quit
```

Using the CPFORMAT EXEC

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

```
==> cpformat ?

Synopsis:

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

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

The following example shows how to attach five 3390-3 volumes and use **CPFORMAT** to format them as paging space. Refer to the planning work sheets that you filled out in 2.7.2, “z/VM DASD worksheet” on page 18:

> **For 3390-9 volumes:** If you are installing onto 3390-9s only one more paging volume is recommended. This will give you two full volumes, or the same equivalent of six 3390-3s. For example if your z/VM system installed onto DASD at A700-A702 and you plan to put the second volume on A703, you would replace `A705-A709` below with **`A703`**.
>
> This will give you enough page space, but can hit performance if all paging requests go to the same volume.

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

```
==> att <a705-a709> *
A705-A709 ATTACHED TO MAINT
==> cpformat <a705-a709> as page

Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A705 MAINT    A705 3390  MPA705 A705          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A706 MAINT    A706 3390  MPA706 A706          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A707 MAINT    A707 3390  MPA707 A707          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A708 MAINT    A708 3390  MPA708 A708          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A709 MAINT    A709 3390  MPA709 A709          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    A705 MAINT    A705 3390  MPA705 A705          0       3339
MAINT    A706 MAINT    A706 3390  MPA706 A706          0       3339
MAINT    A707 MAINT    A707 3390  MPA707 A707          0       3339
MAINT    A708 MAINT    A708 3390  MPA708 A708          0       3339
MAINT    A709 MAINT    A709 3390  MPA709 A709          0       3339
```

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

4.6.2 Formatting DASD for minidisks

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

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

   ```
   LOGON SYSMAINT
   z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
   ```

```
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 06:37:28 EDT MONDAY 11/27/06
z/VM V5.2.0    2006-07-18 08:48


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

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

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

3. Attach the seven volumes that you will use for the controller user ID and the eighth that will be used for the first Linux clone. In this example it is the DASD at addresses A70A-A711. Invoke the **CPFORMAT** command against these volumes using the parameter **as perm**:

   Note: For RHEL5, the example would use DASD addresses A728-A72E.

   ```
   ==> att <a70a-a711> *
   A70A-A711 ATTACHED TO MAINT
   ==> cpformat <a70a-a711> as perm

   Format the following DASD:
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70A MAINT    A70A 3390  MMA70A A70A          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70B MAINT    A70B 3390  MMA70B A70B          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70C MAINT    A70C 3390  MMA70C A70C          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70D MAINT    A70D 3390  MMA70D A70D          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70E MAINT    A70E 3390  MMA70E A70E          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70F MAINT    A70F 3390  MMA70F A70F          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A710 MAINT    A710 3390  MMA710 A710          0       3339
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A711 MAINT    A711 3390  MMA711 A711          0       3339

   WARNING - this will destroy data!
   ARE YOU SURE you want to format the DASD as PERM space (y/n)?
   y
   ...
   DASD status after:
   TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
   MAINT    A70A MAINT    A70A 3390  MMA70A A70A          0       3339
   MAINT    A70B MAINT    A70B 3390  MMA70B A70B          0       3339
   MAINT    A70C MAINT    A70C 3390  MMA70C A70C          0       3339
   MAINT    A70D MAINT    A70D 3390  MMA70D A70D          0       3339
   MAINT    A70E MAINT    A70E 3390  MMA70E A70E          0       3339
   MAINT    A70F MAINT    A70F 3390  MMA70F A70F          0       3339
   MAINT    A710 MAINT    A710 3390  MMA710 A710          0       3339
   MAINT    A711 MAINT    A711 3390  MMA711 A711          0       3339
   ```

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

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

4.6.3 Updating the SYSTEM CONFIG file

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

1. Now that the PAGE and PERM volumes are ready for use, they must be added to the `SYSTEM CONFIG` file so z/VM knows about them. 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    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
```

   It is good to remember this sequence of steps.

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

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

    CP_Owned   Slot   1  520RES
    CP_Owned   Slot   2  520W01
    CP_Owned   Slot   3  520W02
    CP_Owned   Slot   4  520SPL
    CP_Owned   Slot   5  520PAG
    CP_Owned   Slot   6  <MPA705>
    CP_Owned   Slot   7  <MPA706>
    CP_Owned   Slot   8  <MPA707>
    CP_Owned   Slot   9  <MPA708>
    CP_Owned   Slot   10  <MPA709>
    CP_Owned   Slot   11  RESERVED
    CP_Owned   Slot   12  RESERVED
    CP_Owned   Slot   13  RESERVED
...
```

3. 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 via the `User_Volume_Include` statement. If you are using the labelling convention enforced by the **`CPFORMAT EXEC`**, then add the following single line to include all PERM space as volume labels all begin with “`MM`” (the labeling convention used the by the `CPFORMAT EXEC`):

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

```
...
```

4. Save your changes with the **FILE** subcommand. Verify the integrity of the changes with the **CPSYNTAX** command then put the MAINT CF1 disk back online. The following example shows how you did this previously:

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> rel f (det
DASD 0CF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
```

4.6.4 Testing the changes

It is recommended that you again shutdown and reIPL to test the changes. Before you shut down, note that you have only one page volume (520PAG) via 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
------ ---- ---------- ---------- ------ ------ ------ ----
520PAG A702          1       3338 600840     12     12   1%
                                  ------ ------        ----
SUMMARY                           601020     12          1%
USABLE                            601020     12          1%
```

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

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

After the system comes back, **logon as MAINT** and look at the page space again. If you are using 3390-3s, you should see output similar to the following:

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
520PAG A702          1       3338 600840      0      0   0%
MPA705 A705          0       3338 601020      0      0   0%
MPA706 A706          0       3338 601020      0      0   0%
MPA707 A707          0       3338 601020      0      0   0%
MPA708 A708          0       3338 601020      0      0   0%
MPA709 A709          0       3338 601020     12     12   1%
                                  ------ ------        ----
SUMMARY                            3521K     12          1%
USABLE                             3521K     12          1%
```

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

4.7 Creating a user ID for common files

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

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

4.7.1 Define the user in the USER DIRECT file

A small 20 cylinder minidisk is allocated at virtual address 191 and a larger 300 cylinder minidisk (approximately 225MB), 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 18). Cylinder 0 should always be reserved for the label therefore you should start minidisks at cylinder 1.

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

   Note: For RHEL5, the example would use DASD volume `MMA72A`, not `MMA70C`.

   ```
   ==> x user direct c
   ====> bottom
   ====> a 6
   ...
   USER LNXMAINT LNXMAINT 64M 128M BEG        1
    INCLUDE TCPCMSU                           2
    LINK TCPMAINT 592 592 RR                  3
    MDISK 0191 3390 0001 0020 <MMA70C> MR READ WRITE MULTIPLE  4
    MDISK 0192 3390 0021 0300 <MMA70C> MR ALL WRITE MULTIPLE   5
   *                                                           6
   ...
   ====> file
   ```

Note the following points for the numbers in black:

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

2 Include the profile named `TCPCMSU` (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 `MMA70C`

5 Define 192 minidisk of `size 300` cylinders (approximately 225MB) from volume `MMA70C` with the special read password of **`ALL`** which allows read access from any user ID without a disk password

6 An empty comment line for better readability.

2. Whenever an `MDISK` statement is added or modified in the `USER DIRECT` file you should always check for overlapping cylinders and gaps (gaps will only leave empty disk space, however z/VM will allow you to *shoot yourself in the foot* by defining multiple minidisks over the same disk space). This is done with the **`DISKMAP`** command:

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

   File USER DISKMAP A has been created.
   ```

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

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

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

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

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

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

   ```
   ====> /gap
   -----------------------------------------------------------------------

   VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
                                               0         500         501    GAP
   $$$$$$   DATAMOVE    5F0     3380       00501       00501       00001
            DATAMOVE    5FF     3380       00502       00502       00001

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

   VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
                                               0           0           1    GAP
   MMA70C   LNXMAINT   0191     3390       00001       00020       00020
            LNXMAINT   0192     3390       00021       00320       00300
   ...
   ```

   The two GAPs should be listed on the right side: a gap of 501 cylinders on the $$$$$$ volume and a new gap of 1 cylinder exists on the volume that was just used to create disk space for the LNXMAINT user ID; in this case the MMA70C volume.

   Don't worry about the 501 cylinder gap, but to avoid a 1 cylinder gap being reported on each user volume, it is recommended to use the user ID $ALLOC$. This user is set to NOLOG which means it can never be logged onto. Thus it is not a conventional user ID, rather, it is a convenient place to put dummy minidisk definitions for cylinder 0 of all PERM volumes.

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

5. Get out of the file USER DISKMAP with the **QUIT** command or by pressing **F3**.
6. Edit the USER DIRECT file again and add a new minidisk definition:

   ```
   ==> x user direct
   ====> /user $alloc
   USER $ALLOC$  NOLOG
    MDISK A01 3390 000 001 520RES R
    MDISK A02 3390 000 001 520W01 R
    MDISK A03 3390 000 001 520W02 R
    MDISK A04 3390 000 001 <MMA70C> R
   ```

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

   ```
   ==> diskmap user
   ==> x user diskmap
   ====> prefix off
   ====> /$ALLOC
   ...
   VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
   MMA70C   $ALLOC$     A04     3390       00000       00000       00001
            LNXMAINT   0191     3390       00001       00020       00020
            LNXMAINT   0192     3390       00021       00320       00300
   ...
   ```

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

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

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

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

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

4.7.2 Logging and customizing the new user ID

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

1. **Logoff of `MAINT`** and **logon to `LNXMAINT`**.

   ```
   LOGON LNXMAINT
   z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES: 0003 RDR,   NO PRT,   NO PUN
   LOGON AT 05:41:34 EST THURSDAY 01/04/07
   z/VM V5.2.0    2006-07-18 08:48

   DMSACP112S A(191) device error
   ```

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

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

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

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

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

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

4.7.3 Copying a PROFILE XEDIT

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

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

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

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

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

4.7.4 Creating a PROFILE EXEC

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

1. Create the new file and add the following lines. 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` access the `TCPMAINT 592` disk as file mode E, sets CP run on, and sets the retrieve keys per the convention.

2. You could test your changes by logging off and logging back on. However, typing the command **`PROFILE`** will do the same. By default CMS tries to access the 191 disk as A and the 192 disk as D. Also you should have the `TCPMAINT 592` disk accessed as `E`. To see your minidisks, use the **`QUERY DISK`** command:

```
==> profile
DMSACP723I E (592) R/O
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        2          9-01       3591       3600
LXM192 192  D   R/W   300 3390 4096        0         11-00      53989      54000
TCM592 592  E   R/O    67 3390 4096      877       8167-68       3893      12060
MNT190 190  S   R/O   100 3390 4096      689      14325-80       3675      18000
MNT19E 19E  Y/S R/O   250 3390 4096     1010      26665-59      18335      45000
MNT191 120  Z   R/O   175 3390 4096       36        224-01      31276      31500
```

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

```
==> q pf11
PF11 RETRIEVE FORWARD
==> q pf12
PF12 RETRIEVE BACKWARD
```

4.7.5 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. **Log off of LNXMAINT** so that the 192 disk can be accessed read-write.

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

```
# cd /nfs/virt-cookbook-S10/vm    // if you are working with SLES 10
# cd /nfs/virt-cookbook-RH5/vm    // if you are working with RHEL 5
```

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

```
# ftp <129.40.178.120>
220-FTPSERVE IBM VM Level 520 at LAT120.PBM.IHOST.COM, 15:20:32 EDT THURSDAY 2006-11-23
220 Connection will close if idle for more than 5 minutes.
User (129.40.178.120:(none)): lnxmaint
331 Send password please.
Password:
230 LNXMAINT logged in; working directory = LNXMAINT 191
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> prompt
Interactive mode off
ftp> mput *
...
ftp> quit
```

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

```
==> filel * * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=8 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      CHPW52   XEDIT    D1 V         70        180          3  1/04/07  5:50:42
      CPFORMAT EXEC     D1 V         79        231          3  1/04/07  5:50:42
      LABEL520 EXEC     D1 V         75        108          2  1/04/07  5:50:42
      LABEL520 XEDIT    D1 V         71         41          1  1/04/07  5:50:42
      LBL520-9 EXEC     D1 V         75         99          1  1/04/07  5:50:42
      LBL520-9 XEDIT    D1 V         71         43          1  1/04/07  5:50:42
      PROFILE  EXEC     D1 V         63         26          1  1/04/07  5:50:42
      RHEL5    CONF-RH5 D1 V         38         13          1  1/04/07  5:50:42
      RHEL5    EXEC     D1 V         67          9          1  1/04/07  5:50:42
      RHEL5    PARM-RH5 D1 V         80          3          1  1/04/07  5:50:42
      SLES10   EXEC     D1 V         68          9          1  1/04/07  5:50:42
      SLES10   PARM-S10 D1 V         65          9          1  1/04/07  5:50:42
      SLES9X   EXEC     D1 V         74          9          1  1/04/07  5:50:42
      SLES9X   PARMFILE D1 V         73          8          1  1/04/07  5:50:42
      SWAPGEN  EXEC     D1 V         72        358          5  1/04/07  5:50:42
```

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

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

- Configure Linux to shut down gracefully via the **`SET SIGNAL`** command
- Overcommit memory via the **`SET SRM STORBUF`** command
- Grant access to the VSWITCH for each Linux user
- Start user IDs that should be started via the **`XAUTOLOG`** command

- Limit minidisk cache in central storage and turn it off in expanded storage

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

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

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

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

3. Edit the file and add the emboldened text.

```
==> x profile exec
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'                /* start up TCPIP */
'CP XAUTOLOG DTCVSW1'              /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'              /* start VSWITCH controller 2 */
'cp set pf12 ret'                  /* set the retrieve key */
'cp set mdc stor 0m 128m'          /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'          /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 180'       /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */

/* XAUTOLOG each Linux user that should be started */

'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 may not be optimal. For more reading on these values see the following Web sites:
>
> http://www.vm.ibm.com/perf/tips/linuxper.html
> http://www.vm.ibm.com/perf/tips/prgmdcar.html
> http://www.zjournal.com/PDF/robinson.pdf

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

4.8.2 Testing the changes

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

1. Shutdown and reIPL your system.

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

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

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

4. Query the SRM values to see that the new STORBUF settings is in effect and the SIGNAL SHUTDOWN value is set to 180 seconds:

```
==> q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=300% Q2=250% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
...
==> q signal shutdown
System default shutdown signal timeout: 180 seconds
```

This output shows that your changes have taken effect.

4.9 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 Computer Associates VM:Secure. They allow you to address more security issues such as password aging and the auditing of users/ access attempts.

High level z/VM security

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

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

Linux user ID privilege classes

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

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

z/VM user ID and minidisk passwords

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

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

User IDs The password required to logon with

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

Both types of passwords should be modified. This can be done using the CHPW52 XEDIT macro defined 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 a profile named CHPW52 XEDIT has been included with this book. The source code is in Appendix B.1.2, “The CHPW52 XEDIT macro” on page 237.

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

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

1. **Logon to MAINT**.
2. Link and access the LNXMAINT 192 disk to pick up the **CHPW52 EXEC:**

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

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

```
==> copy user direct c = direwrks = (oldd
==> x user direct c
====> /lnx4vm
DMSXDC546E Target not found
====> quit
```

The Target not found message shows that the string LNX4VM is not used in the USER DIRECT file, so it is a good candidate for a password.

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

```
==> x user direct c (profile chpw52) lnx4vm

Changing all passwords to: LNX4VM

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

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

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

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

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

```
HCPDIR494I User directory occupies 39 disk pages
```

Your new directory is online. Do not forget the new password!

Note that this **XEDIT** macro will only work on a vanilla USER DIRECT file because it searches for the original user IDs next to passwords. If you want to change your password again, it should be much easier as you can use the **XEDIT CHANGE** subcommand. For example to change all passwords from lnx4vm to vm5lnx, invoke the following commands:

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

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

4.10 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 520RES, 520W01, 520W02, 520SPL and 520PAG (or just the first three if you are using 3390-9s) You also have configured a sixth volume that is important to Linux: that is the first 320 cylinders of the volume with LNXMAINT on it. Back up that entire volume, because the remainder of it will be used for Linux backup data also.

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

4.11 Relabeling the system volumes

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

When installing z/VM it is possible to modify all but one volume label, that of the 520RES volume. This alleviates the problem that is described next, but it does not alleviate the problem of duplicate volume names.

To understand this possibility, refer to Figure 4-9 on page 62. The z/VM system with the lower device addresses starting at E340 should IPL fine (though you may see a warning at system startup time about duplicate volume labels). However, if the z/VM system starting at device address F000 is IPLed, the 520RES volume will be used, but the remaining volumes in the system are searched for by volume label, not by device address. 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!

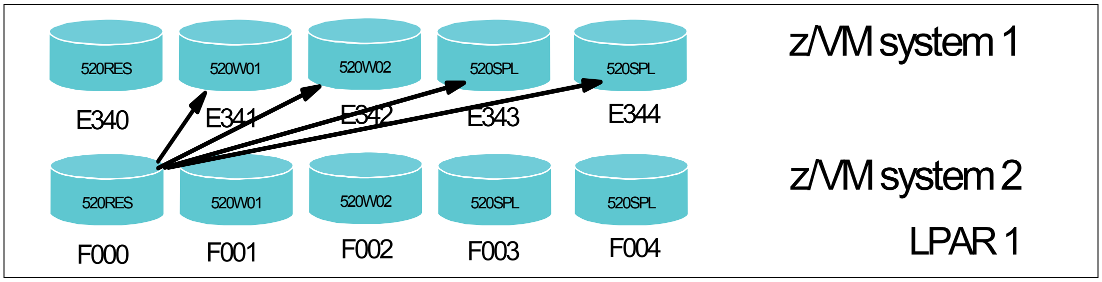


Figure 4-9 The problem with two z/VM systems with identical volume labels

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

- "Modifying labels in the SYSTEM CONFIG file" on page 62
- "Modifying labels in the USER DIRECT file" on page 64
- "Changing the labels on the five volumes" on page 65
- "Shutting down your system and restarting it" on page 66

> **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, don't answer your phone or e-mail, turn off instant messaging :))

4.11.1 Modifying labels in the SYSTEM CONFIG file

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

1. Start an **Integrated 3270 Console** session on the HMC from the *CPC Recovery* (or just *Recovery*) menu.
2. If you have not already done so, **Logon to MAINT** and link and access the `LNXMAINT 192` disk to pick up the **`LABEL520 EXEC`** and **`XEDIT`** macro:

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

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

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  520RES  A700  Own    Online and attached
   2  520SPL  A701  Own    Online and attached
   3  520PAG  A702  Own    Online and attached
   4  520W01  A703  Own    Online and attached
   5  520W02  A704  Own    Online and attached
   6  MPA705  A705  Own    Online and attached
   7  MPA706  A706  Own    Online and attached
   8  MPA707  A707  Own    Online and attached
   9  MPA708  A708  Own    Online and attached
  10  MPA709  A709  Own    Online and attached
  11  ------  ----  -----  Reserved
  12  ------  ----  -----  Reserved
  ...
```

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

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  520RES  9300  Own    Online and attached
   2  520SPL  9301  Own    Online and attached
   3  520PAG  9302  Own    Online and attached
   ...
```

4. The labeling convention described in 2.2.1, “Volume labeling convention” on page 11 suggests using ‘M’ in the second character of the label. An **XEDIT** macro, LABEL520 XEDIT, is supplied to help make this process more reliable. It can be used on both the SYSTEM CONFIG and USER DIRECT files. To modify the labels in the SYSTEM CONFIG file, release the A CP-disk and access it read-write. Back up the SYSTEM CONFIG file:

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

5. Edit the SYSTEM CONFIG file with the **LABEL520 XEDIT** macro passing the five addresses of the z/VM system volumes (a700-a704 in this example):

```
==> x system config f (profile label520) <a700> <a701> <a702> <a703> <a704>
DMSXCG517I 3 occurrence(s) changed on 3 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

For 3390-9s: If z/VM is installed onto 3390-9s, there is a macro named **LBL520-9 XEDIT** that takes only three parameters. For example, if the system volumes are 9300-9302, the command to edit the SYSTEM CONFIG file would be as follows:

```
==> x system config f (profile lbl520-9) <9300> <9301> <9302>
DMSXCG517I 3 occurrence(s) changed on 3 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

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

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

    CP_Owned   Slot   1  MVA700
    CP_Owned   Slot   2  MVA701
    CP_Owned   Slot   3  MVA702
    CP_Owned   Slot   4  MVA703
    CP_Owned   Slot   5  MVA704
    CP_Owned   Slot   6  MPA705
    CP_Owned   Slot   7  MPA706
    CP_Owned   Slot   8  MPA707
    CP_Owned   Slot   9  MPA708
    CP_Owned   Slot   10  MPA709
    CP_Owned   Slot   11  RESERVED
    ...
====> file
```

7. Verify there are no syntax errors:

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

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

```
==> rel f (det
DASD 0CF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
Ready; T=0.01/0.01 09:19:57
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  520RES A700 CKD         39         83
MNTCF2 MAINT    0CF2  B   R/O  520RES A700 CKD         84        128
MNTCF3 MAINT    0CF3  C   R/O  520RES A700 CKD        129        188
```

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

1. Modify the USER DIRECT file again using the **LABEL520 XEDIT** macro. You should see many more occurrences of the labels being changed:

```
==> copy user direct c = direwrks = (oldd rep
==> x user direct c (profile label520) <a700> <a701> <a702> <a703> <a704>
DMSXCG517I 84 occurrence(s) changed on 84 line(s)
DMSXCG517I 134 occurrence(s) changed on 134 line(s)
DMSXCG517I 69 occurrence(s) changed on 69 line(s)
DMSXCG517I 2 occurrence(s) changed on 2 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

For 3390-9s: Again the macro named **LBL520-9 XEDIT** should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to edit the SYSTEM CONFIG file would be as follows:

```
==> x user direct c (profile lbl520-9) <9300> <9301> <9302>
DMSXCG517I 283 occurrence(s) changed on 283 line(s)
DMSXCG517I 2 occurrence(s) changed on 2 line(s)
DMSXCG517I 1 occurrence(s) changed on 1 line(s)
```

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

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

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

4.11.3 Changing the labels on the five volumes

In this section you will change the labels on the 5 volumes via the **CPFMTXA** command. You could do this one volume at a time with the **CPFMTXA LABEL** command. However, the **LABEL520** EXEC has been written to make this step easier.

1. Use LABEL520 with the physical addresses of the 5 system volumes:

```
==> label520 <a700> <a701> <a702> <a703> <a704>
The volumes are:
DASD A700 CP OWNED  520RES   50
DASD A701 CP OWNED  520SPL   1
DASD A702 CP OWNED  520PAG   0
DASD A703 CP OWNED  520W01   93
DASD A704 CP OWNED  520W02   1

The system volume labels will become:
MVA700 MVA701 MVA702 MVA703 MVA704

ARE YOU SURE you want to relabel the DASD (y/n)?
y
HCPCCF6209I INVOKING ICKDSF.
...
ICK03000I CPVOL REPORT FOR 0123 FOLLOWS:

          VOLUME SERIAL NUMBER IS NOW = MVA700
...
VOLUME SERIAL NUMBER IS NOW = MVA701
...
VOLUME SERIAL NUMBER IS NOW = MVA702
...
VOLUME SERIAL NUMBER IS NOW = MVA703
...
VOLUME SERIAL NUMBER IS NOW = MVA704
...
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
DASD 0A03 DETACHED
```

For 3390-9s: The **LBL520-9 EXEC** should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to relabel the system volumes would be as follows:

```
==> lbl520-9 <9300> <9301> <9302>
The volumes are:
DASD 9300 CP OWNED  520RES   142
DASD 9301 CP OWNED  520SPL   2
DASD 9302 CP OWNED  520PAG   0

The system volume labels will become:
MV9300 MV9301 MV9302

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

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

   ```
   ==> directxa user
   z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
   EOJ DIRECTORY UPDATED
   HCPDIR494I User directory occupies 41 disk pages
   Ready(00005); T=0.01/0.01 14:30:37
   ```

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

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

4.11.4 Shutting down your system and restarting it

You will need an HMC console session for this step, if you are not already running from there.

To test the changes you must shut your system down and then restart it. You cannot do a `SHUTDOWN REIPL` in this situation because you will have to do a **FORCE** start

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

Perform the following steps to bring the system back up:

1. From the HMC click the **LOAD** icon in the *CPC Recovery* (or just *Recovery*) menu.
2. Select the **Clear** radio button. All the other parameters should be correct from the previous IPL. Click **OK**
3. Click **Yes** on the *Load Task Confirmation* panel.
4. 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**
5. Press the **F10** key to continue the IPL of your z/VM system. This takes around 3 minutes.
6. At the Start prompt you have to specify a FORCE start, again because the spool volume label has changed. Enter the following:

   ```
   ==> force drain
   ```

7. Do not change the time of day clock:

   ```
   ==> no
   ```

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

   ```
   ==> disc
   ```

You should now be able to get a 3270 emulator session as the TCPIP service machine should be up. Get a 3270 session as MAINT and verify the volume labels have changed with the **QUERY CPOWNED** command:

```
==> q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  MVA700  A700  Own    Online and attached
   2  MVA701  A701  Own    Online and attached
   3  MVA702  A702  Own    Online and attached
   4  MVA703  A703  Own    Online and attached
   5  MVA704  A704  Own    Online and attached
```

```
  6  MPA705  A705  Own    Online and attached
  7  MPA706  A706  Own    Online and attached
  8  MPA707  A707  Own    Online and attached
  9  MPA708  A708  Own    Online and attached
 10  MPA709  A709  Own    Online and attached
 11  ------  ----  -----  Reserved
...
```

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

> **Important:** Only do this if your saved segments have been destroyed! To rebuild saved segments, try the following commands:
>
> ```
> ==> vmfsetup zvm cms
> ==> sampnss cms
> ==> i 190 cl parm savesys cms
> ==> vmfbld ppf segbld esasegs segblist ( all
> ```

4.12 Restoring your z/VM system from tape

It is good to practice to restore a system. You don’t 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. IPLing one of the two will probably find the correct W01, W02, PAG and SPL volumes, but IPLing the other one will probably find the wrong set.

Perform this step only if you successfully completed 4.10, “Backing up your z/VM system to tape” on page 61, and 4.11, “Relabeling the system volumes” on page 61. If you have done both, then the system on tape has volume labels of `520`*`xxx`* and the system on DASD has volume labels `MV`*`yyyy`*. You can restore this system to five other 3390-3s.

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

5

Servicing z/VM

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

--Albert Einstein

This section describes how to apply:

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

Both processes are basically the same.

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

The application of corrective service to z/VM is covered in the *z/VM V5R1 Service Guide and VMSES/E Introduction and Reference*. Both of these documents can be downloaded in PDF format from the following URL:

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

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

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

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

5.1 Applying a PTF

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

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

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

The APAR was assigned the following PTF numbers:

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

There are more details on:

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

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

5.1.1 Determining if a PTF has been applied

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

1. Logon to `MAINT` and issue the **`VMFSETUP`** command to set up minidisks for TCPIP and link to them:

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

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

```
==> vmfinfo

PPF Fileid - Help

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

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

       Options: S - select  V - view

  Option   PPF Fileid
    _      $5654260 PPF      D1
```

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

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

```
PTF/APAR Queries

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

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

    _      Abstract of APAR(s)
```

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

```
Query Output - PTF Status

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

No data found
```

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

5.1.2 Getting service via Internet FTP

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

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

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

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

1. Click on **Support and Downloads** at the top menu.

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

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

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

5.1.3 Downloading the service to z/VM

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

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

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

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

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

   ```
   ==> ftp ptf.boulder.ibm.com
   ftp> <owte8a>
   ftp> <h2q9nep9>
   ftp> cd </390268476/c568411202>
   ftp> ascii
   ftp> get ftp1585.txt
   ...
   ftp> binary f 1024
   ftp> get vlst1585.bin
   ...
   150 Opening BINARY mode data connection for vlst1585.bin (7168 bytes).
   7168 bytes transferred in 0.231 seconds. Transfer rate 31.03 Kbytes/sec.
   ```

```
ftp> get vptf1585.bin
...
551936 bytes transferred in 22.272 seconds. Transfer rate 24.78 Kbytes/sec.
ftp> quit
```

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

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

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

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

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

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

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

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

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

5.1.4 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 256MB virtual machine and you must have the minidisk with a lot of free space - that is what the `MAINT 500` minidisk is for.

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

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

2. ReIPL CMS:

```
==> ipl cms
IPL CMS
z/VM V5.2.0    2006-01-24 13:26
==> Press Enter
```

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

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

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

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

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

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

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

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

```
==> vmfview service
************************************************************************
****             SERVICE              USERID: MAINT               ****
************************************************************************
****            Date: 01/31/06            Time: 12:57:09          ****
************************************************************************
====> F3
```

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

5.1.5 Putting the service into production

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

Important: If you run **PUT2PROD** from a 3270 emulator session, you may lose your connection as the TCPIP service machine may be recycled. Therefore you may want to run this command from a console.

In this example, applying PTF UM31613 *did not* affect the emulator session.

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

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

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

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

Your z/VM system should come back in a few minutes. You may wish to perform the steps in section 5.1.1, “Determining if a PTF has been applied” on page 70 to verify that this PTF has been applied.

5.2 Applying a Recommended Service Upgrade or RSU

Applying an 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, however it assumes you are starting with the RSU tape. Following is an example of upgrading to a z/VM 5.2 RSU with the medium being files, not tape.

The section that follows is a summary of applying service and also describes how to obtain the service via envelope files over the Internet.

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

```
==> q cplevel
z/VM Version 5 Release 2.0, service level 0501 (64-bit)
Generated at 11/18/05 23:44:09 EST
IPL at 12/19/06 13:44:37 EST
```

The service level four digits are split in half, where the first half is the last two digits of the year and the second half is the service level for that year. Therefore, a service level of “0501” means the first service level for the year 2005 has been applied. If you installed from DVD, that was on the second RSU DVD disc.

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

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

5.2.1 Getting service from the Internet

The PTF number for the most current RSU for z/VM 5.2.0 is **UM97520** The PTF number for the most current RSU for z/VM 5.1.0 is UM97510. Point a Web browser to:

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

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

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

2. Click on **Support and Downloads** at the top menu.

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

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

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

5.2.2 Downloading the service to z/VM

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

```
TEXT    = Data sent via "INET". To retrieve your service:
TEXT    =   FTP to: ptf.boulder.ibm.com
TEXT    =   Log on using userid "owte8a" and password "h2q9nep9"
TEXT    =   Enter the following FTP commands:
TEXT    =     cd /390268476/c568411202
TEXT    =     ascii
TEXT    =     get ftp8476.txt
TEXT    =     binary f 1024
TEXT    =     get rlst8476.bin
TEXT    =     get rptf0176.bin
TEXT    =     get rptf0276.bin
TEXT    =     get rptf0376.bin
```

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

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

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

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

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

```
==> ftp ptf.boulder.ibm.com
ftp> <owte8a>
ftp> <h2q9nep9>
ftp> cd </390268476/c568411202>
ftp> ascii
ftp> get ftp8476.txt
ftp> binary f 1024
ftp> get rlst8476.bin rlst8476.servlink
...
10240 bytes transferred in 0.523 seconds. Transfer rate 19.58 Kbytes/sec
ftp> get rptf0176.bin rptf0176.servlink
...
36944896 bytes transferred in 191.632 seconds. Transfer rate 192.79 Kbytes/sec.
ftp> get rptf0276.bin rptf0276.servlink
...
26028032 bytes transferred in 132.353 seconds. Transfer rate 196.66 Kbytes/sec.
ftp> get rptf0376.bin rptf0376.servlink
...
52193280 bytes transferred in 269.094 seconds. Transfer rate 193.96 Kbytes/sec.
ftp> quit
```

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

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

```
==> deterse rlst8476 servlink a = = = (replace
==> deterse rptf0176 servlink a = = = (replace
==> deterse rptf0276 servlink a = = = (replace
==> deterse rptf0376 servlink a = = = (replace
```

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

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

```
==> browse ftp8476 txt
VM RSU Package Information
--------------------------
This file contains byte counts of files to receive and instructions
for preparing the files for installation. The byte counts listed
below should match the byte counts of the files when they are received
using FTP.

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

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

5.2.3 Receiving, applying, and building the service

You must receive, apply, and build the service. Then it can be put into production.

In the past, this was a cumbersome procedure. For example, to receive, apply and build the CP component, the following steps were needed:

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

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

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

   ```
   ==> def stor 256M
   STORAGE = 256M
   Storage cleared - system reset.
   ==> ipl cms
   IPL CMS
   z/VM V5.2.0    2006-01-24 13:26
   ==> Press Enter
   ```

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

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

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

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

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

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

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

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

   ```
   ==> vmfview service
   ************************************************************************
   ****             SERVICE              USERID: MAINT                ****
   ************************************************************************
   ****            Date: 11/10/05            Time: 11:43:15           ****
   ************************************************************************
   CK:VMFSUI2104I PTF UM30896 contains user information. Review the :UMEMO
   CK:            section in file UM30896 $PTFPART
   CK:VMFSUI2104I PTF UM31044 contains user information. Review the :UMEMO
   CK:            section in file UM31044 $PTFPART
   CK:VMFSUI2104I PTF UM31233 contains user information. Review the :UMEMO
   CK:            section in file UM31233 $PTFPART
   ```

```
CK:VMFSUI2104I PTF UM31275 contains user information. Review the :UMEMO
CK:            section in file UM31275 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (I) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTX EXEBIN
WN:VMFSRV1221W The CP Stand-Alone Dump Utility must be rebuilt. Follow
WN:            the instructions in the z/VM Service Guide.
```

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

You should see that the service was installed successfully.

5.2.4 Putting the service into production

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

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

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

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

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

```
==> q cplevel
z/VM Version 5 Release 2.0, service level 0501 (64-bit)
Generated at 11/18/05 23:44:09 EST
IPL at 12/19/06 13:44:37 EST
```

To invoke the new CP load module, use the **SHUTDOWN REIPL** command. When your system comes back up, it should be at the new CP service level, in this example `0602`:

```
==> shutdown reipl iplparms cons=sysc
...
==> q cplevel
z/VM Version 5 Release 2.0, service level 0602 (64-bit)
Generated at 09/18/06 18:39:52 EST
IPL at 12/19/06 14:53:28 EST
```

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

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/
```

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

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

This shows that the latest APAR applied is VM63895.

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

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

This shows the controller is named DTCVSW1. Then use the **NETSTAT** command with the controller name to determine the maintenance of the TCPIP MODULE:

```
==> netstat tcp <dtcvsw1> level
VM TCP/IP Netstat Level 520
IBM 2094; z/VM Version 5 Release 2.0, service level 0601 (64-bit), VM TCP/IP Lev
el 520; RSU 0601 running TCPIP MODULE E2 dated 02/10/06 at 19:00
TCP/IP Module Load Address:  00C44000
```

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

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

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.

Part 2

RHEL 5

This part of the book describes how to set up installation, customization and cloning of RHEL 5 servers. Included is a chapter on using kickstart, Red Hat's automated install procedure.

6

Configuring an NFS server for RHEL 5

“*Anyone who has never made a mistake has never tried anything new.*”

— Albert Einstein

A common method of installing Linux on under z/VM is over the network from another server using the Network File System (NFS). To accomplish this, a PC Linux system is recommended. This server supplies both the RHEL 5 distribution and the files associated with this book. The server must have at least 4 GB of free disk space. It can be a Linux PC, but it can also be a UNIX box (Sun™ Solaris™, Hewlett Packard HP-UX, IBM AIX® or other). You can also choose to use a Windows workstation via FTP or HTTP. Often, more problems are encountered when using a Windows workstation than a Linux or Unix workstation to serve the RHEL 5 install tree.

The steps in this chapter explain how to configure a PC Linux box as the NFS server. Red Hat Installation Guide for the IBM S/390 and IBM System z Architectures manual provides additional information about the installation options on the Web at:

```
http://www.redhat.com/docs/manuals/enterprise/
```

The following tasks will set up a Linux NFS server:

- “Installing Linux on the PC” on page 83
- “Downloading files associated with this book” on page 84
- “Setting up a RHEL 5 install tree” on page 84
- “Enabling the NFS server” on page 86

6.1 Installing Linux on the PC

If you don’t have a Linux PC then you must get access to one on the network and install Linux onto it. Describing that is outside the scope of this book. However, installing the same distribution onto a PC server that you plan to install on System z is recommended. Doing so will give you practice with the installation process and will give you a reference system that may be helpful in understanding the differences between the Intel (i386, i686) and System z (s390x) architctures. In this chapter, a PC running RHEL 5 is used.

6.2 Downloading files associated with this book

This book has files associated with it to make the task of customizing, cloning and backup up your virtual servers easier. The tar file on the Web at:

```
http://linuxvm.org/present/misc/virt-cookbook-RH5.tgz
```

The tar file `virt-cookbook-RH5.tgz` is about 124KB. Download the file and untar it. The following example shows this being done from a newly created directory `/nfs/`:

```
# mkdir /nfs
# cd /nfs
... download or copy the file virt-cookbook-RH5.tgz to /nfs/ ...
# tar xzf virt-cookbook-RH5.tgz
```

List the files in the new directory `virt-cookbook-RH5/`:

```
# cd virt-cookbook-RH5
# ls
README.txt   sitar-1.0.7-270.1.noarch.rpm        vm/
rsyncd.conf  virtcook-controller-1-RH5.s390x.rpm
```

The `README.txt` file briefly describes each of the files and the one directory. You may want to briefly view that file. You now have downloaded and uncompressed the files associated with this book.

6.3 Setting up a RHEL 5 install tree

You must have a valid Red Hat entitlement for Linux on IBM System z to access the Red Hat Enterprise Linux 5 ISO images. If you do not have one, you can request a free evaluation copy at:

```
http://www.redhat.com/rhel/details/eval/
```

Follow the link named Red Hat Enterprise Linux AS for IBM eServer zSeries and IBM S/390 and create an account. You require this login to access the Red Hat Network (RHN), where you can download the installation discs at:

```
https://rhn.redhat.com
```

RHEL 5 is distributed on physical CDs or files that are ISO images of CDs. This is not new. What is new is that it is also distributed on a single physical DVD disc or a single ISO image. It is easier to work with a single DVD ISO image than to work with multiple CD ISO images, so this approach is recommended

Download the DVD ISO image to a new directory `/nfs/rhel5/`:

```
# cd /nfs
... download the DVD ISO image
```

6.3.1 Copying from physical DVD

In the event that you have a physical DVD, but not an ISO image, it is recommended that you create an ISO image. You could skip creating the ISO image and copy the data directly from the DVD to the install tree, but creating the ISO image is recommended so you have a reference file.

Be sure your PC has a DVD drive, not just a CD drive (if you have a PC that only has a CD drive, you can create ISO images of the CDs, but this is not described). Put the DVD in the tray and use the **dd** command to create the ISO image. The device file named /dev/cdrom is often associated with the CD/DVD drive, however, your device file name may be different. If so, you must determine the correct name.

Perform these steps only if you are starting with a physical DVD disc:

```
# cd /nfs
# dd if=/dev/cdrom of=RHEL5-Server-s390x-DVD.iso
# umount /mnt/cdrom
```

You should now have an ISO image of the DVD.

6.3.2 Verifying the ISO image

An important early step is to verify the integrity of DVD ISO image. This is done by comparing a checksum value which was calculated when the DVD was created against a checksum value calculated against your ISO image. If the two checksum values differ then there was an error somewhere in the copying process.

The **md5sum** command allows you to compare checksum files. The checksum value for the GA version of RHEL 5 for the s390x architecture is as follows:

```
# cat MD5SUM
c15890d1aa9addb285dad89fb8ade1db  RHEL5-Server-s390x-DVD.iso
```

Run the **md5sum** command against the MD5SUM file:

```
# md5sum -c MD5SUM
RHEL5-Server-s390x-DVD.iso: OK
```

Important: Your MD5SUM file may have checksum values for the DVD *and* the CD ISO images. If this is true and you only have one DVD ISO image, the md5sum will generate errors of the form:

```
md5sum: RHEL5-Server-s390x-disc1-ftp.iso: No such file or directory
RHEL5-Server-s390x-disc1-ftp.iso: FAILED open or read
```

This is not a problem, as long as the DVD ISO image is reported as OK.

If the ISO image does not report OK, it must be downloaded or copied again until it does.

6.3.3 Copying the DVD contents

Copy the contents of the ISO image to the file system. Temporarily mount it over a new directory tmp/ using a loopback device:

```
# cd /nfs
# mkdir tmp
# mount -o loop RHEL5-Server-s390x-DVD.iso tmp
```

List the contents of the mounted ISO image:

```
# ls tmp
EULA            README-pa.html         RELEASE-NOTES-ml.html
eula.en_US      README-pt_BR.html      RELEASE-NOTES-mr.html
generic.ins     README-ru.html         RELEASE-NOTES-or.html
...
```

Make a new directory, /nfs/rhel5/, and recursively copy the contents of the DVD to it via the **cp -a** command. This will take a number of minutes to complete:

```
# cp -a tmp/* rhel5/
```

6.3.4 Building the repository directory

For the yum command to work, a common metadata repository must be built via the **createrepo** command. There is a sample repository in the directory `Server/repodata/`. The group XML file named `comps-rhel5-server-core.xml` should be used to create group information i

```
# cd rhel5/Server/
# mv repodata/ repodata.orig
# createrepo -g repodata.orig/comps-rhel5-server-core.xml .
2495/2495 - junit-javadoc-3.8.2-3jpp.1.s390x.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

Now the newly created `repodata/` directory contains the correct common medata:

```
# cd repodata
# ls
comps-rhel5-server-core.xml  other.xml.gz    repomd.xml
```

6.4 Enabling the NFS server

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

- Export the appropriate directories.
- Start the NFS server in the current run level.

The directories to export via NFS are set in the `/etc/exports` configuration file. Export the directory `/nfs/rhel5/` to make the install tree available and `/nfs/virt-cookbook-RH5/` to make the files associated with this book available. First make a backup copy of the file. Then edit the original copy and add the two directories as follows:

```
# cd /etc
# cp exports exports.orig
# vi exports       // add two lines
/nfs/rhel5            *(ro,sync)
/nfs/virt-cookbook-RH5 *(ro,sync)
```

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

Set the NFS server to start via the **chkconfig** command and start it on for the current session with the **service nfs start** command:

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

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

```
# mkdir /mnt/tmp
# mount localhost:/nfs/rhel5/ /mnt/tmp
# ls -F /mnt/tmp
EULA            README-or.html         RELEASE-NOTES-ja.html
eula.en_US      README-pa.html         RELEASE-NOTES-ko.html
generic.ins     README-pt_BR.html      RELEASE-NOTES-ml.html
GPL             README-ru.html         RELEASE-NOTES-mr.html
images/         README-si.html         RELEASE-NOTES-or.html
...
```

This shows that the RHEL 5 install tree is accessible via NFS. Now unmount it and test the virt-cookbook-RH5/ directory:

```
# umount /mnt/tmp
# mount localhost:/nfs/virt-cookbook-RH5 /mnt/tmp
# ls -F /mnt/tmp
README.txt  sitar-1.0.7-270.1.noarch.rpm         vm/
clone.sh*   virtcook-controller-1-RH5.s390x.rpm
# umount /mnt/tmp
```

You should now be able to use this server as the source of a RHEL 5 mainframe Linux installation. Later you will be able to copy the install tree to a System z Linux virtual server.

7

Installing and configuring RHEL 5

"*The most incomprehensible thing about the world is that it is at all comprehensible.*"

— Albert Einstein

In this chapter you will install your first Linux ID, RHEL5. This Linux ID is unique in that it will have two copies of Linux; think of it as a dual-boot Linux PC. You will install the following Linux images onto the `RHEL5` user ID:

Master image — The copy of Linux that will be cloned. This has also been refered to as the *Golden image*. This should be as lean as possible so as to be a generic virtual server and to fit comfortably in one 3390-3 DASD.

Controller — The copy of Linux that will normally be running and one that does the cloning. This has also been refered to as *the cloner*.

- In addition to being *the cloner*, the RHEL5 user ID serves as the administration point for current and future Linux IDs
- RHEL 5 install server - a tree of RPMs and other files needed for installation
- NFS server - to export the *install tree* and possibly other data and directories
- Kickstart server: This hosts files necessary for automated installations.
- Backup server - for incremental backup of key virtual server configuration files
- Time server - for time synchronization of all other Linux virtual servers

In this section, you will perform following tasks:

- "Creating the user ID RHEL5" on page 90
- "Adding RHEL5 to AUTOLOG1's PROFILE EXEC" on page 92
- "Preparing RHEL5 bootstrap files" on page 93
- "Installing the master image" on page 95
- "Configuring the master image" on page 105
- "Installing the controller" on page 116
- "Configuring the controller" on page 121

Chapters 4, 5 and 6 must be completed before proceeding.

7.1 Creating the user ID RHEL5

In this section you will define the RHEL5 user ID to z/VM.

1. **Logon to** MAINT and edit the USER DIRECT file:

   ```
   ==> x user direct c
   ```

   In the USER DIRECT file you can group statements that will be common to many user definitions in a construct called a *profile*. This profile can then become part of the user definitions via the INCLUDE statement. You used the existing profile TCPCMSU when you defined the LNXMAINT user.

2. Create a new profile named LNXDFLT. This will contain the user directory statements that will be common to all Linux user IDs. To save typing, you can use the **""** prefix commands to duplicate the IBMDFLT profile that should be on lines 37-50:

   ```
   ""037 ****************************************************************
   00038 *
   00039 PROFILE IBMDFLT
   00040   SPOOL 000C 2540 READER *
   00041   SPOOL 000D 2540 PUNCH A
   00042   SPOOL 000E 1403 A
   00043   CONSOLE 009 3215 T
   00044   LINK MAINT 0190 0190 RR
   00045   LINK MAINT 019D 019D RR
   00046   LINK MAINT 019E 019E RR
   00047   LINK MAINT 0402 0402 RR
   00048   LINK MAINT 0401 0401 RR
   00049   LINK MAINT 0405 0405 RR
   ""050 ******************************
   ```

3. Edit the duplicated profile by deleting the three LINK MAINT 040x lines, and inserting the lines that are emboldened:

   ```
   PROFILE LNXDFLT
     IPL CMS                                  1
     MACHINE ESA 4                            2
     CPU 00 BASE                              3
     CPU 01                                   4
     NICDEF 600 TYPE QDIO LAN SYSTEM VSW1     5
     SPOOL 000C 2540 READER *
     SPOOL 000D 2540 PUNCH A
     SPOOL 000E 1403 A
     CONSOLE 009 3215 T
     LINK MAINT 0190 0190 RR
     LINK MAINT 019D 019D RR
     LINK MAINT 019E 019E RR
     LINK LNXMAINT 192 191 RR                 6
     LINK TCPMAINT 592 592 RR                 7
   ```

   Notes:

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

4. Go to the bottom of the file and add the definition for a new user ID named RHEL5. This user ID is given class B privilege, aside from the typical class G, 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 (MMA728-MMA72E in this example) with the labels of your DASD:

```
USER RHEL5 LNX4VM 256M 1G BDEG
  INCLUDE LNXDFLT
  OPTION LNKNOPAS APPLMON
  MDISK 100 3390 0001 3338 <MMA728> MR LNX4VM LNX4VM LNX4VM
  MDISK 200 3390 0001 3338 <MMA729> MR LNX4VM LNX4VM LNX4VM
  MDISK 203 3390 0321 3018 <MMA72A> MR LNX4VM LNX4VM LNX4VM
  MDISK 204 3390 0001 3338 <MMA72B> MR LNX4VM LNX4VM LNX4VM
  MDISK 205 3390 0001 3338 <MMA72C> MR LNX4VM LNX4VM LNX4VM
  MDISK 206 3390 0001 3338 <MMA72D> MR LNX4VM LNX4VM LNX4VM
  MDISK 207 3390 0001 3338 <MMA72E> MR LNX4VM LNX4VM LNX4VM
*
```

This Linux user ID will have the following minidisks and virtual disks (VDISKs):

Table 7-1 Minidisks to be defined

Minidisk	Description
100	The root file system of the master image - this is the copy that will be cloned.
101-102	These are virtual disk (VDISK) swap spaces that are not defined in USER DIRECT file, but defined by calls to the **SWAPGEN EXEC** in the user's **PROFILE EXEC** so that when the user ID logs on the VDISKs are created.
200	The root file system of the Linux controller. This will serve as the administration point for all your Linux virtual servers.
203	The /backup/ file system for backing up. Note that the first 320 cylinders of the volume used for this disk were used for the LNXMAINT 191 and 192 minidisks.
204-207	Minidisks used to create a logical volume mounted over /nfs/ for making the RHEL5 installation tree and the files associated with this book available via NFS.

5. Go back to the top of the file and search for string USER $ALLOC$. Add cylinder 0 of each of the six new volumes to this dummy user ID so they don't show up as gaps in the USER DISKMAP report file (in this example MMA72A was added earlier)

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
  MDISK A01 3390 000 001 <MVA700> R
  MDISK A02 3390 000 001 <MVA703> R
  MDISK A03 3390 000 001 <MVA704> R
  MDISK A01 3390 000 001 <MVA700> R
  MDISK A02 3390 000 001 <MVA703> R
  MDISK A03 3390 000 001 <MVA704> R
  MDISK A04 3390 000 001 <MMA72A> R
  MDISK A05 3390 000 001 <MMA728> R
  MDISK A06 3390 000 001 <MMA729> R
  MDISK A07 3390 000 001 <MMA72B> R
  MDISK A08 3390 000 001 <MMA72C> R
  MDISK A09 3390 000 001 <MMA72D> R
  MDISK A0A 3390 000 001 <MMA72E> R
...
====> file
```

6. Run **DISKMAP** to check for overlaps and gaps. You should only see the single 501 cylinder gap.

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

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

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

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

7.2 Adding RHEL5 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 the RHEL5 user ID to the sections that grant access to the VSWITCH and that **XAUTOLOG** 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 DTCVSW1'               /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'               /* start VSWITCH controller 2 */
'cp set pf12 ret'                   /* set the retrieve key */
'cp set mdc stor 0m 128m'           /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'           /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 180'        /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant rhel5'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog rhel5'

'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 rhel5
Command complete
```

7.3 Preparing RHEL5 bootstrap files

To IPL a RHEL5 installation system, four bootstrap files must be prepared. Three are punched to z/VM reader and then IPLed. These three files IPLed are a kernel, a parameter file and an initial RAMdisk. The fourth file is a configuration file stored on a CMS disk that the parameter file points to.

Think of these as the files that are on as a PC Linux boot CD (or floppy disk). Also, a small REXX EXEC is commonly used to clean out the reader, punch the three files and IPL the reader. A sample RHEL5 parameter file, configuration file and RHEL5 EXEC are supplied and should be on the LNXMAINT 192 disk. This is described in section 7.1, "Creating the user ID RHEL5" on page 90. Therefore, only the kernel and RAMdisk should need to be copied.

1. Start an SSH session as root on the NFS server.
2. Use the **ftp** command to copy the RHEL 5 kernel and initial RAMdisk to LNXMAINT's D disk. These files must have a record format of fixed 80 byte records. This format can be set with the **site fix 80** FTP subcommand (if this subcommand fails, try **quote site fix 80**). Following is an example:

```
# cd /nfs/rhel5/images
# ftp <129.40.178.120>
Name (129.40.178.120:root): lnxmaint
Password:
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> site fix 80
200 Site command was accepted.
ftp> bin
200 Representation type is IMAGE.
ftp> put initrd.img rhel5.initrd
12237790 bytes sent in 01:28 (135.35 KB/s)
ftp> put kernel.img rhel5.kernel
local: kernel.img remote: rhel5.kernel
3629384 bytes sent in 00:25 (138.96 KB/s)
ftp> quit
```

3. Go back to your 3270 session. **Logoff of MAINT and logon to LNXMAINT**.
4. The files RHEL5 EXEC, RHEL5 PARM-RH5 and RHEL5 CONF-RH5 should exist on the LNXMAINT 192 disk (D) as they were copied in 7.1, "Creating the user ID RHEL5" on page 90. Use the **FILELIST** command to verify that they were copied in Fixed 80 byte record format. You should see the following files (the number of records and blocks may vary):

```
==> filel rhel5 * 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
      RHEL5    KERNEL   D1 F         80      45368        832  1/15/07 10:25:20
      RHEL5    INITRD   D1 F         80     153196       2993  1/15/07 10:24:47
      RHEL5    CONF-RH5 D1 V         38         13          1  1/15/07  9:45:41
      RHEL5    EXEC     D1 V         67          9          1  1/15/07  9:45:41
      RHEL5    PARM-RH5 D1 V         80          3          1  1/15/07  9:45:41
```

5. Quit by pressing **F3**.

6. Verify that the file **RHEL5 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 rhel5 exec d
   /* EXEC to punch RHEL5 install system to reader and IPL from it */
   'CP SPOOL PUN *'
   'CP CLOSE RDR'
   'PUR RDR ALL'
   'PUN RHEL5       KERNEL * (NOH'
   'PUN' userid() 'PARM-RH5 * (NOH'
   'PUN RHEL5       INITRD * (NOH'
   'CH RDR ALL KEEP'
   'IPL 00C CLEAR'
   ```

7. There are two text files needed to install RHEL5: a parameter file and a configuration file. The parameter file is named RHEL5 PARM-RH5 and is punched to the reader. It has some values, the most important value, the CMSCONFFILE variable, points to the configuration file which remains on a CMS minidisk. The sample parameter file should not need to be modified:

   ```
   ==> type rhel5 parm-rh5 d
   ramdisk_size=40000 root=/dev/ram0 ro ip=off
   CMSDASD=191 CMSCONFFILE=RHEL5.CONF-RH5
   vnc vncpassword=lnx4vm
   ```

8. Enter the appropriate networking values in the configuration file, in this case RHEL5 CONF-RH5. The sample configuration file contains some fields that should be correct and many fields that will have to be modified. Modify at least the fields shown in **<bold>** and perhaps others. Refer to the worksheet in section 2.7.3, “Linux resources worksheet” on page 19.

   Before:

   ```
   DASD=100-102,200,203-207
   HOSTNAME=hostName.DNSname.com
   NETTYPE=qeth
   IPADDR=n.n.n.n
   SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
   NETWORK=n.n.n.n
   NETMASK=255.255.255.0
   SEARCHDNS=DNSname.com
   BROADCAST=n.n.n.n
   GATEWAY=n.n.n.n
   DNS=n.n.n.n
   MTU=1500
   PORTNAME=DONTCARE
   ```

   After:

   ```
   ==> x rhel5 conf-rh5
   DASD=100-102,200,203-207
   HOSTNAME=<lat128.pbm.ihost.com>
   NETTYPE=qeth
   IPADDR=<129.40.178.128>
   SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
   NETWORK=<129.40.178.0>
   NETMASK=255.255.255.0
   SEARCHDNS=<pbm.ihost.com>
   BROADCAST=<129.40.178.255>
   GATEWAY=<129.40.178.254>
   DNS=<129.40.106.1>
   MTU=1500
   ```

```
PORTNAME=DONTCARE
```

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

Now you are ready to start the master image installation.

7.4 Installing the master image

The master or golden image will be installed onto the RHEL5 100 disk. Again, this is the image of Linux that will be cloned.

You should still be logged onto LNXMAINT. When you logon to the new RHEL5 user ID, you will pick up a PROFILE EXEC from LNXMAINT 192. This file runs when you press **Enter** at the VM READ prompt and creates two VDISKs via the SWAPGEN EXEC to later be used as swap spaces. View the contents of the PROFILE EXEC via the CMS **TYPE** command:

```
==> type profile exec d
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 101 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 102 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (id = 'SLES9'|id = 'SLES9X'|id = 'SLES10'|id = 'RHEL5') then
do
  iplDisk = 200
end
else /* this is a Linux clone */
do
  iplDisk = 100
end
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL' iplDisk
else  /* user is interactive -> prompt */
do
  say 'Do you want to IPL Linux from DASD' iplDisk'? y/n'
  parse upper pull answer .
  if (answer = 'Y') then
    'CP IPL' iplDisk
end /* else */
```

Perform the following steps to prepare for installing Linux:

1. **Logoff of LNXMAINT and logon to RHEL5**. When you logon, you should see message indicating that a virtual NIC has been created at addresses 600-602 (via the **NICDEF** statement in the USER DIRECT file) and that two VDISKs have been created at virtual addresses 101 and 102 (via the **SWAPGEN EXEC** called from the **PROFILE EXEC**):

```
LOGON RHEL5
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0602 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 10:00:35 EST WEDNESDAY 12/20/06
z/VM V5.2.0    2006-12-19 14:40
```

```
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (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 DASD 200? y/n
==> n
```

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

```
==> q v 100-102
q v 100-102
00: DASD 0100 3390 MMA728 R/W       3338 CYL ON DASD  A728 SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 0013
00: DASD 0102 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 0014
```

4. Verify that you have a virtual OSA at addresses 600-602 with the **QUERY VIRTUAL OSA** command:

```
==> q v osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0007
00:      0600 DEVTYPE OSA        CHPID 0E OSD
00:      0600 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
00: OSA  0601 ON NIC  0600  UNIT 001 SUBCHANNEL = 0008
00:      0601 DEVTYPE OSA        CHPID 0E OSD
00:      0601 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
00: OSA  0602 ON NIC  0600  UNIT 002 SUBCHANNEL = 0009
00:      0602 DEVTYPE OSA        CHPID 0E OSD
00:      0602 QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
```

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

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

6. This is adequate memory to run a RHEL5 Linux image, however it is often too small in which to install Linux. Temporarily modify the storage up to 512MB with the **DEFINE STORAGE** command. Then **IPL CMS** and again answer **n** to the question of IPLing Linux:

```
==> def stor 512m
00: STORAGE = 512M
00: Storage cleared - system reset.
==> ipl cms
z/VM V5.2.0    2006-07-18 08:48

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 200? y/n
n
```

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

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

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

7.4.1 Stage 1 of the RHEL 5 installation

Follow these steps to begin the installation of RHEL5.

1. Run the **RHEL5 EXEC**. You should see many screens of questions and answers scrolling by. If you had used the default parameter file shipped with RHEL5, you would have had to answer all the networking questions manually. With the proper parameters set in RHEL5 CONF-RH5, the install process should proceed to where you have to use a browser to VNC client get into the installation program:

```
==> RHEL5
   00: RDR FILE 0019 SENT FROM RHEL5    PUN WAS 0019 RECS 045K CPY  001 A NOHOLD NO
   KEEP
   00: RDR FILE 0020 SENT FROM RHEL5    PUN WAS 0020 RECS 0003 CPY  001 A NOHOLD NO
   KEEP
   00: RDR FILE 0021 SENT FROM RHEL5    PUN WAS 0021 RECS 153K CPY  001 A NOHOLD NO
   KEEP
   00: 0000003 FILES CHANGED
   Linux version 2.6.18-1.2839.el5 (brewbuilder@devel8.z900.redhat.com) (gcc versio
   n 4.1.1 20061130 (Red Hat 4.1.1-43)) #1 SMP Tue Dec 5 22:58:42 EST 2006
   We are running under VM (64 bit mode)
   Detected 2 CPU's
   Boot cpu address  0
   Built 1 zonelists.  Total pages: 131072
   Kernel command line: ramdisk_size=40000 root=/dev/ram0 ro ip=off
                        CMSDASD=191 CMSCONFFILE=RHEL5.CONF-RH5
                        vnc vncpassword=lnx4vm
   ...
```

   The kernel should continue to boot until you see the following messages:

```
   ...
   Starting telnetd and sshd to allow login over the network.

   Connect now to 129.40.178.128 to start the installation.
```

2. From your workstation, use your SSH client (e.g. PuTTY) to connect to the IP address and begin the installation. When prompted for a user name, enter **root**: a password will not be required. Figure 7-1 shows the initial screen of the installer. Use the **Tab** key to move between fields. Use the arrow keys to move among choices and **Enter** to select a choice.

Important: If the DASD you are using has never been formatted for Linux, you may get many screens of warning messages similar to the following on your 3270 session:

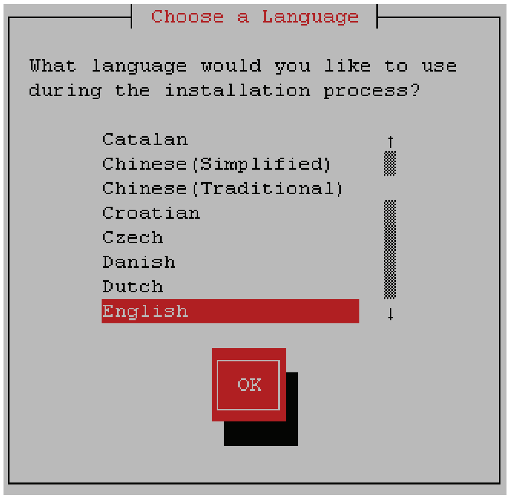
```
dasd(eckd): I/O status report for device 0.0.0100:
dasd(eckd): in req: 000000000e027ee8 CS: 0x40 DS: 0x0E
dasd(eckd): device 0.0.0100: Failing CCW: 000000000e027fd0
dasd(eckd): Sense(hex)  0- 7: 00 08 00 00 04 ff ff 00
```

This is not a problem, you just have to clear the screen many times or the install process will freeze. An alternative to clearing the screen many times is to issue the following CP **TERM** command:

```
#cp term more 0 0
```

Press **Enter** and the screen should scroll freely. The downside of this option is that you may miss some messages that are important. You may later want to set the value back to the default of waiting 50 seconds to beep then another 10 seconds to clear the screen via the following command:

```
#cp term more 50 10
```

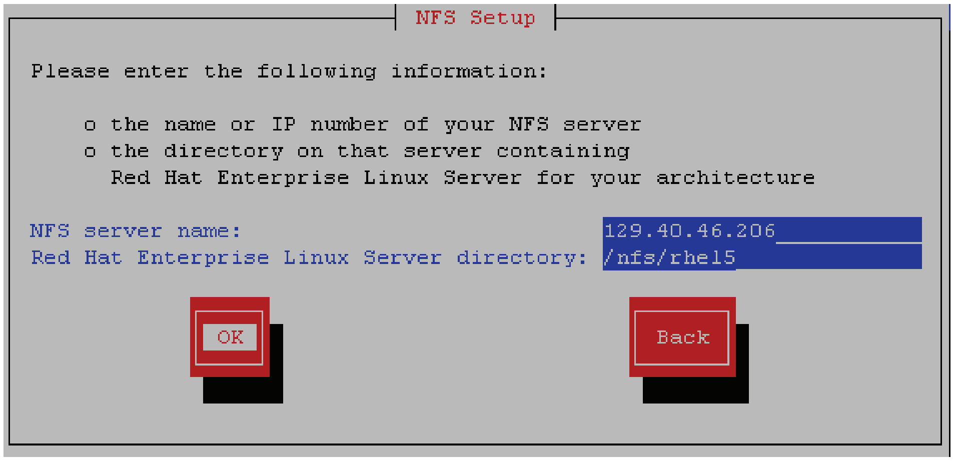


Choose a Language

What language would you like to use during the installation process?

Figure 7-1 Initial screen of installer

3. The `Choose a Language` screen should appear. Select your language, Tab to **OK** and press **Enter**.
4. The *Installation Method* screen should appear. Choose **NFS image** for the install method, and select **OK**.
5. The *NFS Setup* screen should appear. Enter the IP address of the PC NFS server on the first line, then the path to the installation tree on the second line, and select **OK**. See the example in Figure 7-2 which uses the NFS server at IP address 129.40.46.206:



NFS Setup

Please enter the following information:

o the name or IP number of your NFS server
o the directory on that server containing
  Red Hat Enterprise Linux Server for your architecture

NFS server name: 129.40.46.206
Red Hat Enterprise Linux Server directory: /nfs/rhel5

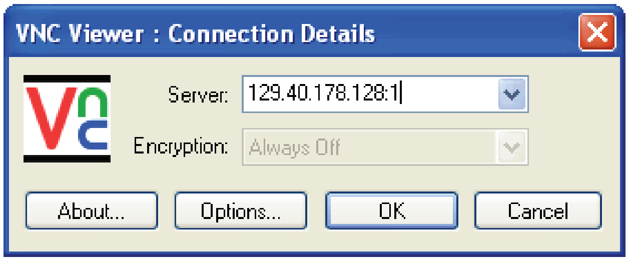
Figure 7-2 NFS setup screen

6. Now the curses Windows should end and the install program (anaconda) should start a VNC server. You should see messages similar to the following:

```
Welcome to the anaconda install environment 1.1 for zSeries

Running anaconda, the Red Hat Enterprise Linux Server system installer - please
wait...
Starting VNC...
The VNC server is now running.
Please connect to 129.40.178.128:1 to begin the install...
Starting graphical installation...
```

```
Press <enter> for a shell
XKB extension not present on :1
```

7. Start a VNC client (e.g. RealVNC) and connect to the server with your IP address with a **:1** appended to the end as shown in Figure 7-3. When prompted for a password, enter the password specified in the `RHEL5 PARM-RH5` file (**lnx4vm** in the sample file). In the following example, Linux is being installed with the IP address 129.40.178.128:



VNC Viewer : Connection Details

Server: 129.40.178.128:1

Encryption: Always Off

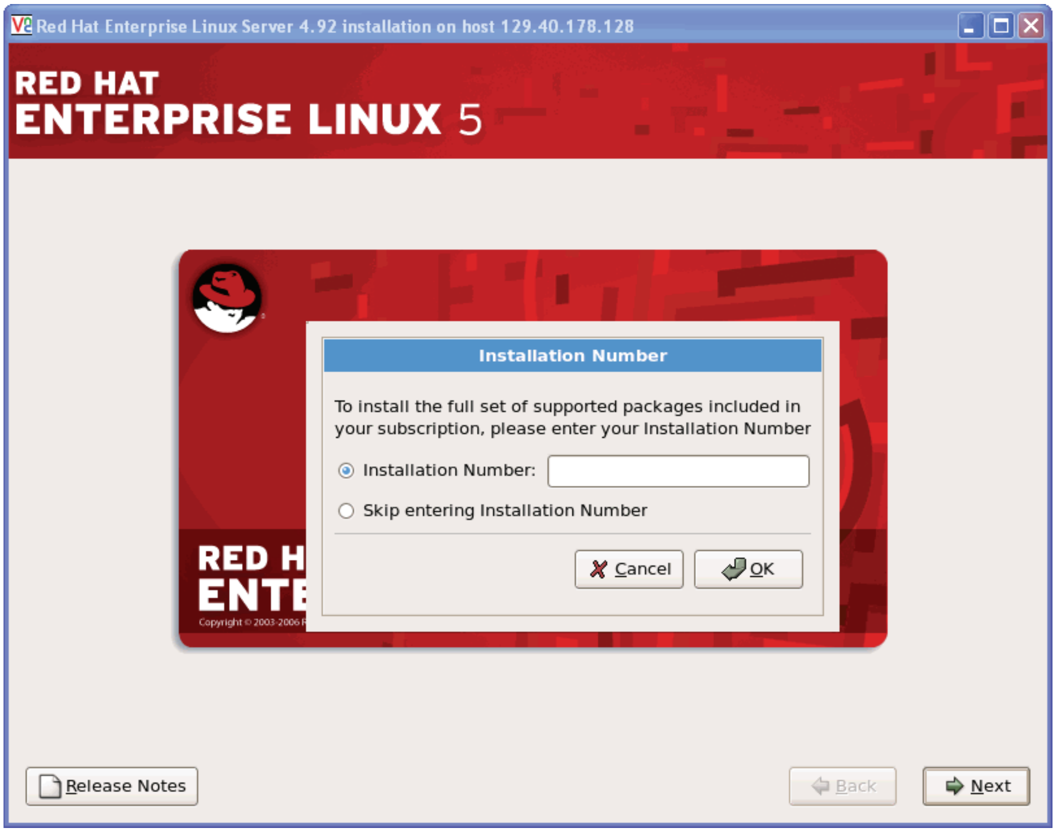
Figure 7-3 Connecting with VNC client

7.4.2 Stage 2 of the RHEL 5 installation

After you have connected using VNC, perform the following steps:

1. The welcome screen opens. Click **`Next`**.
2. The installer prompts for an installation number as shown in Figure 7-4, which determines the packages that are available for installation. This information should have come with the details of your subscription. If you select **`Skip`**, this will result in a core server installation, and additional functionality can be added manually after installation.

   At the time of this writing, the installation number is listed in the *Installation-Related Notes* section of the release notes in the install tree. For English, see `/nfs/rhel5/RELEASE-NOTES-en` on the PC NFS server.



Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.128
RED HAT
ENTERPRISE LINUX 5
RED H
ENTE
Copyright © 2003-2006 R

Installation Number

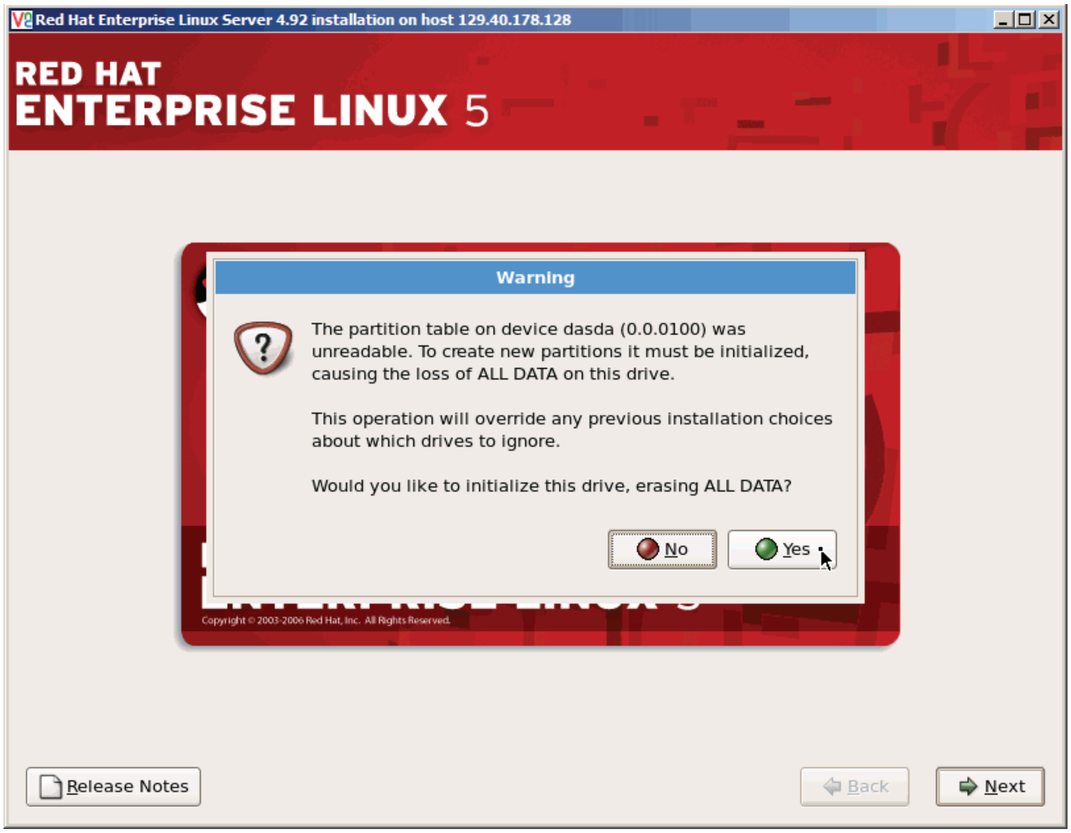
To install the full set of supported packages included in your subscription, please enter your Installation Number

● Installation Number:

○ Skip entering Installation Number

Figure 7-4 Enter installation number

3. Because the disks were just formatted by CPFMTXA (with the CPFORMAT EXEC wrapper), you will be prompted to format each of them. Click **yes** to format dasda (minidisk 100) as shown in Figure 7-5 on page 101. A progress indicator will be shown. You will not be asked to format disks 101 and 102 because they are the VDISK swap spaces and are not recognized by the installer.


Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.128
RED HAT
ENTERPRISE LINUX 5
Warning

The partition table on device dasda (0.0.0100) was unreadable. To create new partitions it must be initialized, causing the loss of ALL DATA on this drive.

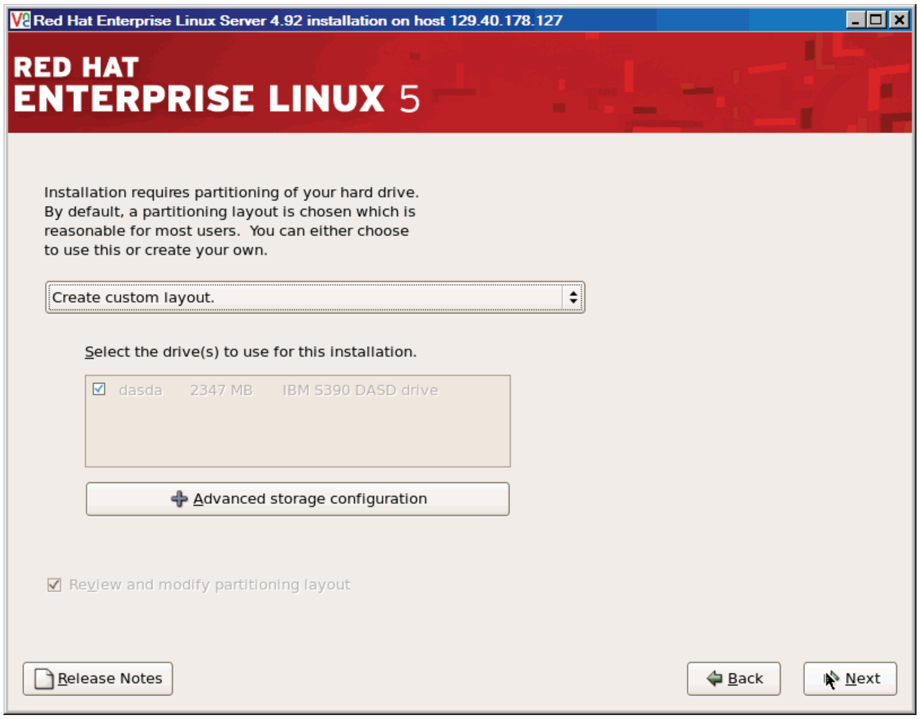
This operation will override any previous installation choices about which drives to ignore.

Would you like to initialize this drive, erasing ALL DATA?

No
Yes
Copyright © 2003-2006 Red Hat, Inc. All Rights Reserved.
Release Notes
Back
Next

Figure 7-5 Formatting DASD for Linux use

4. You have the choice of formatting the dasdd-dasdh disks (203-207 minidisks) now or when you later install the controller. When the installer prompts you to initialize the drive, select **Yes** :
5. The installer now searches for a previous installation. If this is the first installation, nothing will be found. If a previous installation is found, the installer prompts you whether to **Upgrade** or **Install**. Choose **Install** here.
6. At the disk partitioning setup screen, choose **Create Custom Layout** as shown in Figure 7-6 on page 102. Click **Next** to proceed to the Disk Setup screen.


Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.127

RED HAT
ENTERPRISE LINUX 5

Installation requires partitioning of your hard drive.
By default, a partitioning layout is chosen which is
reasonable for most users. You can either choose
to use this or create your own.

Create custom layout.

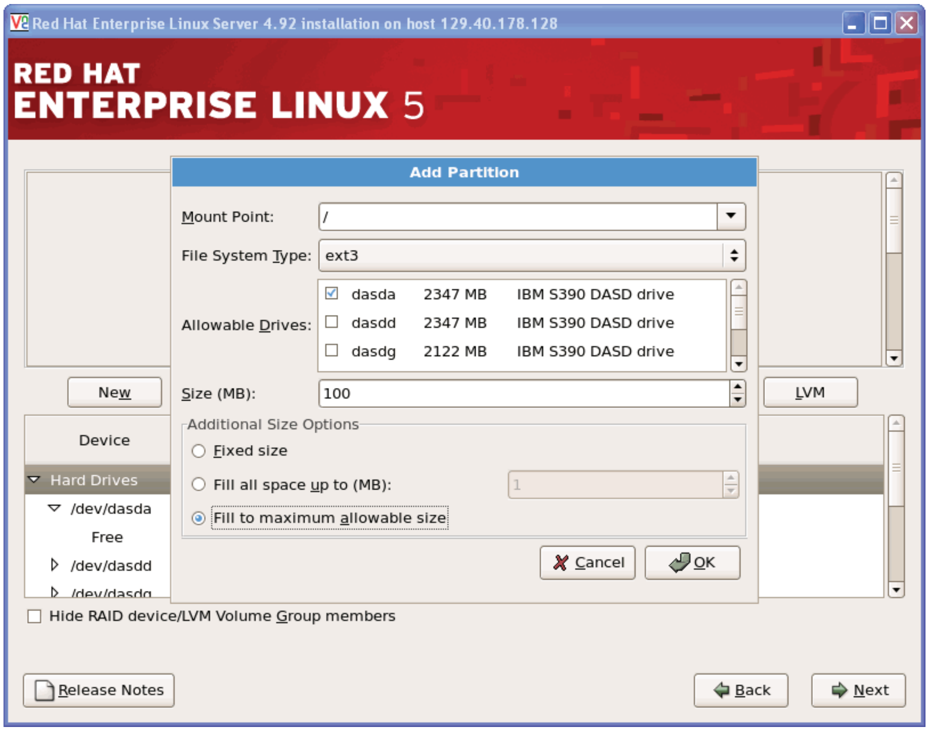
Select the drive(s) to use for this installation.

☑	dasda	2347 MB	IBM S390 DASD drive

☑ Review and modify partitioning layout

Figure 7-6 Choosing custom disk partitioning

7. Disks 100 should show free space. Click the **New** button to add a root file system to the 100 disk. Enter / in the field *Mount Point*, choose to format the file system as ext3 and deselect all drives *except* dasda as Figure 7-7 shows. Finally, select **Fill to maximum allowable size** and click **OK**.


Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.128

RED HAT
ENTERPRISE LINUX 5

Add Partition

Mount Point: /

File System Type: ext3

Allowable Drives:

☑	dasda	2347 MB	IBM S390 DASD drive
□	dasdd	2347 MB	IBM S390 DASD drive
□	dasdg	2122 MB	IBM S390 DASD drive

Size (MB): 100

Additional Size Options

○ Fixed size

○ Fill all space up to (MB): 1

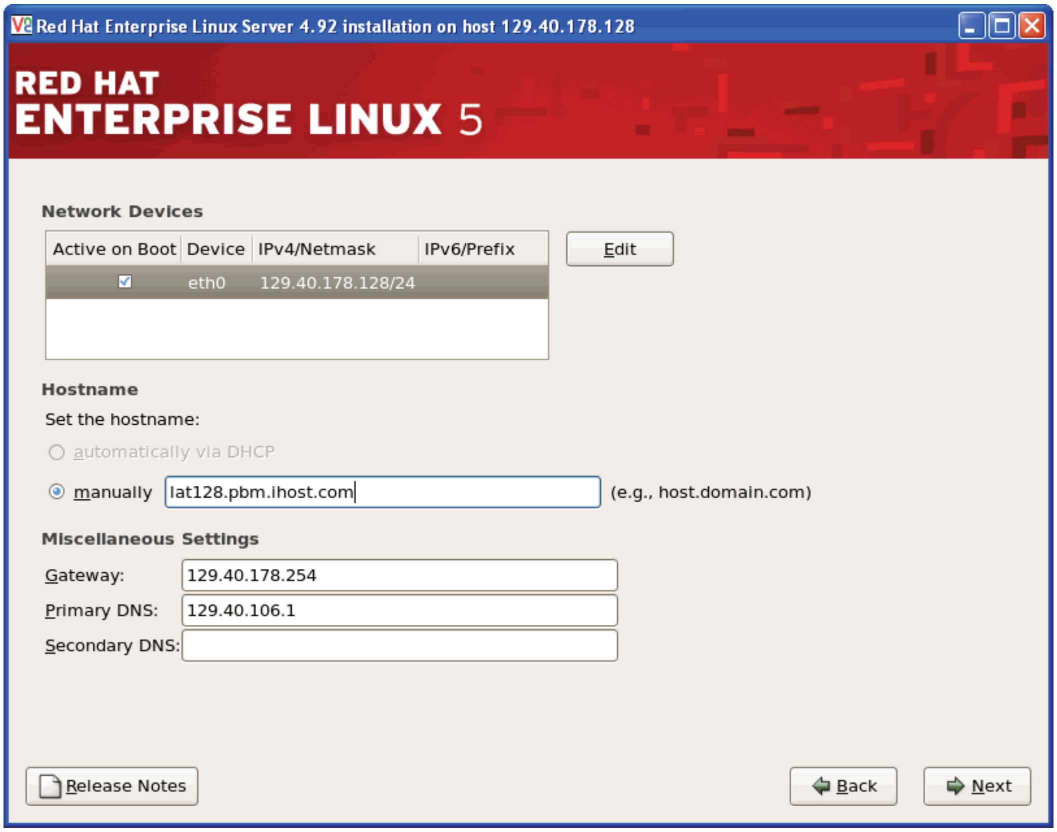
● Fill to maximum allowable size

□ Hide RAID device/LVM Volume Group members

Figure 7-7 Adding the root file system

8. Click **Next**. You will see a warning about missing swap partitions. You can ignore it because there will be swap disk defined later on. Click **Yes**.

9. At the Network Configuration screen, enter a host name for the master image as Figure 7-8 shows. Other network settings have been automatically taken from the `RHEL5 CONF-RH5` file. Click **Next**.



Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.128

RED HAT
ENTERPRISE LINUX 5

Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
☑	eth0	129.40.178.128/24	

Hostname

Set the hostname:

○ automatically via DHCP

● manually lat128.pbm.ihost.com (e.g., host.domain.com)

Miscellaneous Settings

Gateway: 129.40.178.254

Primary DNS: 129.40.106.1

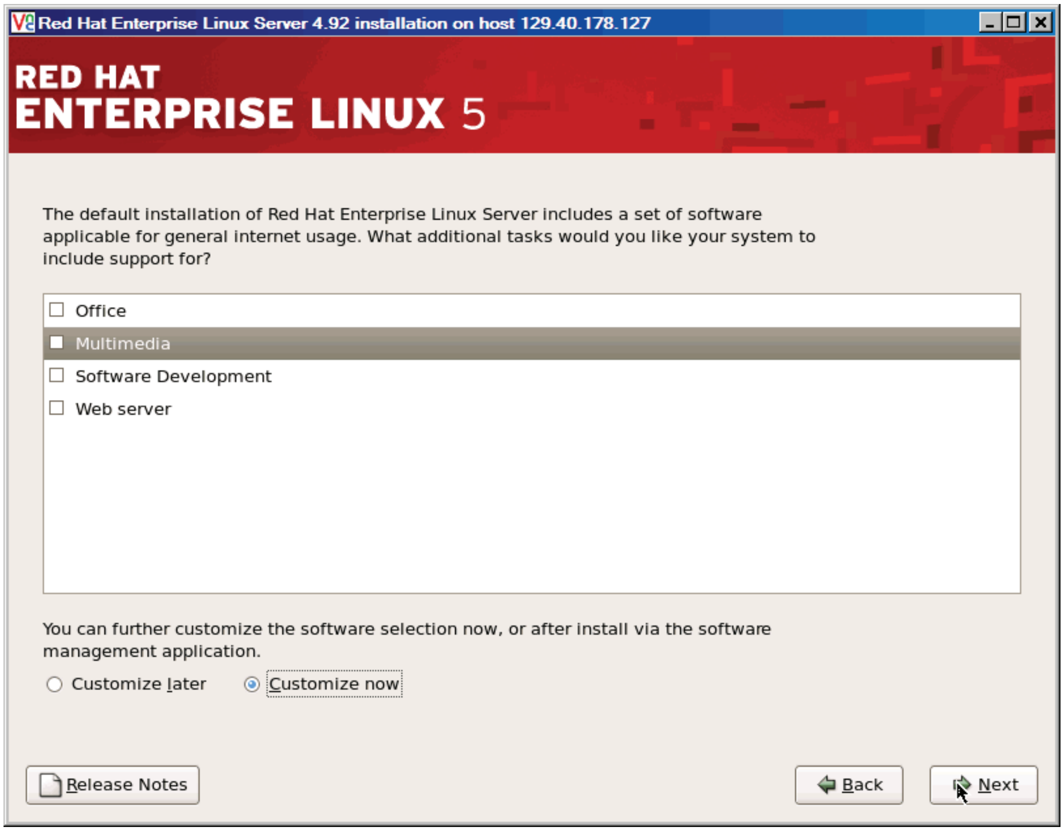
Secondary DNS:

Figure 7-8 Configuring the network device

10. At the Time Zone Selection screen, highlight the nearest city in your time zone. Deselect **System clock uses UTC** and click **Next**.

11. Set the Root Password, then click **Next**.

12. At the package selection screen, deselect all package groups, select `Customize Now` and click `Next`.


Red Hat Enterprise Linux Server 4.92 installation on host 129.40.178.127
RED HAT
ENTERPRISE LINUX 5

The default installation of Red Hat Enterprise Linux Server includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?

☐ Office
☐ Multimedia
☐ Software Development
☐ Web server

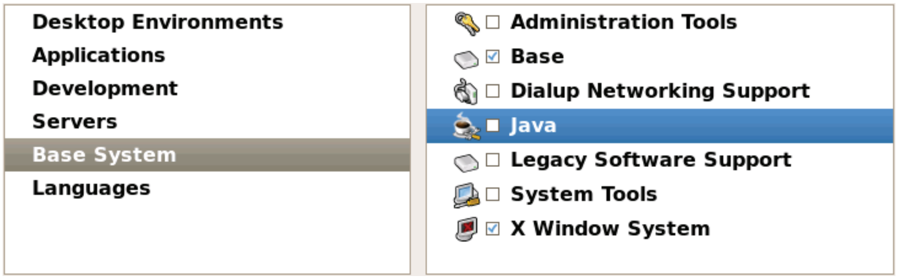
You can further customize the software selection now, or after install via the software management application.

○ Customize later ● Customize now

Figure 7-9 Customizing software packages

13. Remove the majority of the package groups as follows:

   a. Under **Desktop Environments**, deselect **Gnome** so nothing is selected
   b. Under **Applications**, deselect all package groups
   c. Under **Development**, nothing should be selected
   d. Under **Servers**, deselect **Printing Support** so nothing is selected
   e. Under **Base Sysetm**, leave **Base** and **X Window System** selected and deselect everything else, as shown in Figure 7-10.
   f. Under **Languages**, select a package group if you need specific language support.
   g. Click **Next** and package dependencies will be determined.



☐ Administration Tools
☑ Base
☐ Dialup Networking Support
☐ Java
☐ Legacy Software Support
☐ System Tools
☑ X Window System

Figure 7-10 Customizing packages

14. Finally, click **Next** to begin the installation. This lasts from five to ten minutes, depending on network speeds.

15. When the installation is complete, click **Reboot**. The VNC session will end, but your system will not reboot.

7.4.3 Booting your new Linux system from disk

A minimal system should now be installed onto minidisk 100. Return to your z/VM 3270 session and IPL the newly installed system via the command **#CP IPL 100**.

```
 /mnt/sysimage/dev done
 /mnt/sysimage done
you may safely reboot your system
==> #cp ipl 100
CP IPL 100
zIPL v1.3.2 interactive boot menu
  0. default (linux)
  1. linux
Note: VM users please use '#cp vi vmsg <input>'
Please choose (default will boot in 15 seconds):
...
```

Linux will boot after 15 seconds if you take no action. To boot immediately, issue the following command:

```
==> #cp vi vmsg 0
```

You system should continue to boot until a login prompt is presented.

Important: Due to a known issue when connecting to a layer 3 VSWITCH, you must log in through the console and change one networking parameter before the system will come up on the network. The need to do this is explained in the following Web page:

http://kbase.redhat.com/faq/FAQ_85_6247.shtm

Login as root. Invoke the **echo** command to append the **ARP=no** directive to the end of the eth0 interface configuration file, then reboot the Linux server.

```
Red Hat Enterprise Linux Server release 4.92 (Tikanga)
Kernel 2.6.18-1.2839.el5 on an s390x

lat128 login: root
Password:

Last login: Tue Dec 12 11:19:47 from 9.56.60.217
# echo ARP=no >> /etc/sysconfig/network-scripts/ifcfg-eth0
# reboot
```

When your system comes back up, it should be on the network. Disconnect from the 3270 session with the following command:

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

Start an SSH session into the master image as root.

7.5 Configuring the master image

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

7.5.1 Configuring VDISK swap

Recall that you were warned during the installation that you were not using any swap spaces. This can be confirmed with the **swapon -s** command which lists swap spaces:

```
# swapon -s
```

No output means there are no swap spaces. Linux must now be configured to use the VDISK swap spaces. Observe the first three DASD devices via the following **head** command:

```
# head -3 /proc/dasd/devices
0.0.0100(ECKD) at ( 94: 0) is dasda : active at blocksize: 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at ( 94: 4) is dasdb : active at blocksize: 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at ( 94: 8) is dasdc : active at blocksize: 512, 1048576 blocks, 512 MB
```

Of course /dev/dasda is your 100 minidisk and /dev/dasda1 is the first partition which is mounted over the root file system. You should also see that /dev/dasdb and /dev/dasdc are FBA devices which signifies VDISKs. The **SWAPGEN EXEC** creates a single partition with a swap signature on each of the two VDISKs. Therefore, you can use /dev/dasdb1 and /dev/dasdc1 as swap devices. Make a backup copy of the file system table, /etc/fstab, then add the two swap spaces as follows:

```
# cd /etc
# cp fstab fstab.orig
# vi /etc/fstab
LABEL=/1                /                       ext3    defaults        1 1
/dev/dasdb1             swap                    swap    defaults        0 0
/dev/dasdc1             swap                    swap    defaults        0 0
LABEL=/backup           /backup                 ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
/dev/VolGroup00/LogVol00 /nfs                    ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
LABEL=/                 /rhel5master            ext3    defaults        1 2
sysfs                   /sys                    sysfs   defaults        0 0
```

Save the file and turn on the swap spaces via the **swapon -a** command which will read the /etc/fstab file and then verify that the swap spaces have been activated:

```
# swapon -a
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdb1                             partition       259956  0       -1
/dev/dasdc1                             partition       519924  0       -2
```

7.5.2 Configuring automount of install tree

You will now configure the Linux automount servie to mount the installation tree on demand. The automounter will automatically mount a remote directory when it is accessed, and automatically unmount it after a period of inactivity.

Make a backup copy of the file /etc/auto.master, then add the following line at the bottom:

```
# cd /etc
# cp auto.master auto.master.orig
# vi /etc/auto.master   // add one line at the bottom
...
#
+auto.master
/nfs    /etc/auto.controller
```

The new line specifies that the file system mounted beneath the directory /nfs/ will be configured in the file /etc/auto.controller. Now create the file /etc/auto.controller, and add one line which points to the RHEL 5 install tree that is exported:

```
# vi /etc/auto.controller
rhel5           -ro,hard,intr   <129.40.46.206>:/nfs/rhel5
```

This line specifies that beneath /nfs/ (in auto.master) when the directory rhel5/ (field 1) is accessed, automatically used the specified options (in field 2) to mount the specified directory (in field 3).

Now, create the /nfs/ directory. Restart the **autofs** service to pick up the new configuration. Then list the contents of the /nfs/rhel5/ directory. Even though it does not exist as a local file system, it is automatically mounted when referenced:

```
# mkdir /nfs
# service autofs restart
Stopping automount: [  OK  ]
Starting automount: [  OK  ]
# ls /nfs/rhel5
EULA            README-or.html         RELEASE-NOTES-ja.html
eula.en_US      README-pa.html         RELEASE-NOTES-ko.html
...
```

7.5.3 Configuring yum for online updates

You will now configure **yum** so it can install RPMs from automounted install tree. Later you will move the install tree from the PC server to the System z controller. Create a file named rhel5.repo in the /etc/yum.repos.d directory:

```
# cd /etc/yum.repos.d
# vi rhel5.repo
[RHEL5]
name=Red Hat Enterprise Linux 5
baseurl=file:///nfs/rhel5/Server
```

You must import the RPM GPG key so that **yum** knows you are installing official Red Hat packages. The Red Hat GPG key is located in the install tree. Import the key with the following command:

```
# rpm --import /nfs/rhel5/RPM-GPG-KEY
Note: for RHEL5 beta distributions, use the file RPM-GPG-KEY-redhat-beta
```

Note: Red Hat signs each RPM with a private GPG key, which is compared to your public key each time a package is installed. This method ensures that the RPM is a genuine, unaltered package. When installing an RPM, if you ever see a message similar to:

```
Header V3 DSA signature: NOKEY, key ID 897da07a
```

Either the correct GPG key has not been imported, or the package itself has been altered by a third party.

You are now ready to use yum to install or upgrade an RPM package. To install a package, use **yum install <packagename>**. Yum will conveniently install the packages specified and automatically resolve dependencies for you. Note that you should not specify the package version on the command line, only the package name.

7.5.4 Adding additional RPMs

Now that **yum** is configured, you will install some additional RPM packages onto the master image. You may choose to add additional RPMs, or may choose to omit some of the following:

openmotif A window manager used in conjunction with VNC that is more usable than the default window manager

xinetd This is a service that controls access to other services. It is recommended for rsync configuration in “Configuring rsyncd” on page 111.

sitar A tool to generate an HTML file describing the Linux system in great detail.

To add the first two RPMs described above you can use **yum -y install** command packages. Notice that yum has added multiple dependencies to the package list:

```
# yum -y install openmotif xinetd
Loading "rhnplugin" plugin
Loading "installonlyn" plugin
...
Installed: openmotif.s390x 0:2.3.0-0.3.el5 openmotif.s390 0:2.3.0-0.3.el5 xinetd.s390x
2:2.3.14-10.el5
Dependency Installed: expat.s390 0:1.95.8-8.2.1 fontconfig.s390 0:2.4.1-5.el5
freetype.s390 0:2.2.1-16.el5 libXft.s390 0:2.1.10-1.1 libXmu.s390 0:1.0.2-5 libXp.s390
0:1.0.0-8 libXp.s390x 0:1.0.0-8 libXrender.s390 0:0.9.1-3.1 libjpeg.s390 0:6b-37
libpng.s390 2:1.2.10-7
Complete!
```

Adding the Sitar package

The sitar package is not part of RHEL 5, but is an important of the backup and monitoring sections of this book. Therefore it is included in the tar file which is associated with this book. That file should be untarred on the PC NFS server. First mount the directory over /mnt/:

```
# mount <129.40.46.206>:/nfs/virt-cookbook-RH5 /mnt
# cd /mnt
# rpm -Uvh sitar-1.0.7-270.1.noarch.rpm
Preparing...                ########################################### [100%]
   1:sitar                  ########################################### [100%]
```

Now you have a good set of packages in the master image that will be cloned into virtual servers. You may choose to add more or remove some packages.

7.5.5 Changing automount to point to controller

Now that you have installed the necessary additional RPMs, you will configure the master image to automatically mount the controller's NFS share, so that **yum** will use this instead of the PC NFS server. Although the controller has not yet been installed, you will make the change now so that this change is reflected when you clone new virtual servers based on this master image. Edit the file /etc/auto.controller, and change the IP address from the PC NFS server to the controller. In this example, the IP address was changed from 129.40.46.206 to 129.40.178.128

```
# vi /etc/auto.controller
rhel5           -ro,hard,intr   <129.40.178.128>:/nfs/rhel5
```

7.5.6 Turning off unneeded services

There are a number of services which are started in a RHEL 5 minimum system. In order to keep the master image as lean as possible, some of these can be turned off:

Turn off the following services via the **chkconfig** command:

```
# chkconfig cups off
# chkconfig iptables off
# chkconfig ip6tables off
# chkconfig auditd off
# chkconfig haldaemon off
# chkconfig atd off
# chkconfig kudzu off
# chkconfig mdmonitor off
# chkconfig rpcgssd off
# chkconfig rpcidmapd off
# chkconfig anacron off
# chkconfig mcstrans off
# chkconfig yum-updatesd off
```

Note: You should only disable the **iptables** service if you are on a trusted network. Otherwise, you will need to configure **iptables** to allow network traffic for the VNC server and NFS, as well as any other services that require network access.

For more information on configuring **iptables** for NFS traffic, see the article located at:

http://www.redhat.com/magazine/010aug05/departments/tips_tricks/

You may choose to leave these services on, or turn others off. You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig --list | grep 3:on
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
avahi-daemon    0:off   1:off   2:off   3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
messagebus      0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
nfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
readahead_early 0:off   1:off   2:on    3:on    4:on    5:on    6:off
restorecond     0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

```
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
xfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
yum-updatesd    0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

7.5.7 Configuring the VNC server

Often applications require a graphical environment. The Virtual Network Computing (VNC) server allows for a graphical environment to be set up easily by starting the **vncserver** service. RHEL 5 configures the VNC server via the /etc/sysconfig/vncservers configuration file. Add a line at the bottom of this file to specify the VNC user:

```
# vi /etc/sysconfig/vncservers
...
VNCSERVERS="1:root"
```

Set a VNC password via the **vncpasswd** command. This password will be needed to connect to the VNC server:

```
# vncpasswd
Password: <lnx4vm>
Verify: <lnx4vm>
```

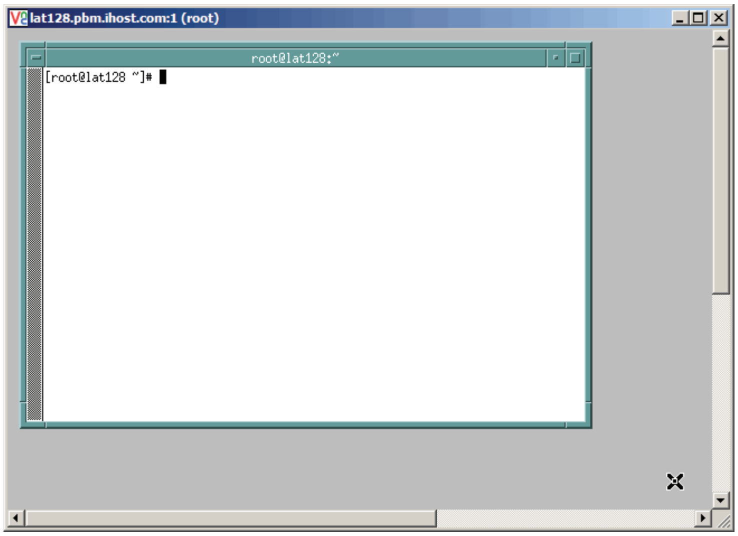
Now start the VNC server with the **service** command:

```
# service vncserver start
Starting VNC server: 1:root xauth:  creating new authority file /root/.Xauthority
New 'lat128.pbm.ihost.com:1 (root)' desktop is lat128.pbm.ihost.com:1
Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/lat128.pbm.ihost.com:1.log
[  OK  ]
```

There is one more configuration to be done. Stop the VNC server and change from the Tiny window manger, **twm**, to the Motif window manager, **mwm**:

```
# cd /root/.vnc
# vi xstartup   // change last line
...
xsetroot -solid grey
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

You can now use the VNC client to connect to the IP address of the master image with a **:1** appended. A sample session is shown in Figure 7-11.



```
[root@lat128 ~]#
```

Figure 7-11 VNC client session to the VNC server

Note that the VNC server will not be started across reboots. When you need a graphical environment, you can either to start the **vncserver** process manually (recommended), or you can use **chkconfig** to enable automatic startup.

7.5.8 Configuring rsyncd

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

First use the **chkconfig** command to enable **rsync** startup at boot time:

```
# chkconfig rsync on
```

You will configure two files: the main configuration file /etc/rsyncd.conf, and a file to allow only the controller to connect, /etc/trusted.hosts.

For convenience, a copy of rsyncd.conf has been included in the tar file associated with this book. You can either create the rsyncd.conf file manually or you can copy it. The following example shows it being copied from the PC NFS server which should still be mounted over /mnt:

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

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

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

```
# vi trusted.hosts // add one line - the TCP/IP address of the controller
<129.40.178.128>
```

Now rsync should be configured to allow the /etc/ directory to be copied by the controller.

7.5.9 Creating the script nightly.sh

Create a small script named **nightly.sh** in the /etc/cron.daily/ directory that will be run each night.

```
# cd /etc/cron.daily
# vi nightly.sh
let secs=$RANDOM/9                    # number of seconds between 0 and 1 hour
sleep $secs                           # sleep so cron load is dispersed
sitar --format=html --outfile=/etc/sitar.html  # create sitar report in /etc
/usr/sbin/ntpd -q                     # Adjust the clock
# chmod +x nightly.sh
```

The first line creates a random number between 0 and 3640 - approximately the number of seconds in one hour. The second line puts the script to sleep for that long so the cron workload across the LPAR does not all start at once.

The third line creates an HTML report about the virtual server in the /etc/ directory via **sitar**. This report will be copied to the controller each night via it's cron script, and then will be referenced by the DAZL Web application

The last line adjusts the clock accurately. In the next section you will set up NTP to point to the controller as a software clock reference.

7.5.10 Setting the software clock accurately

It is important to have the Linux software clock set properly. This is commonly accomplished by running the **ntpd** daemon against some accurate time sources. However, this is expensive in terms of CPU costs. A compromise between an extremely accurate clock (running **ntpd** as a service) and using fewer CPU cycles (not running it) is to reset the clock once a day. This can be done with a script that runs the **ntpd -q** command from the /etc/cron.daily/ directory. NTP will be configured so the controller will be the external clock. In 7.7.9, "Turning on the NTP server" on page 125, the controller will be configured to point to external clocks on the Internet.

Modify the /etc/ntp.conf file to comment out the NTP servers on the Internet and add a pointer to the controller as follows:

```
# cd /etc
# vi ntp.conf   // comment 3 servers and add controller
...

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
# server 0.rhel.pool.ntp.org
# server 1.rhel.pool.ntp.org
# server 2.rhel.pool.ntp.org
server 129.40.178.128
...
```

Now you should have the configuration file /etc/ntp.conf pointing to the controller, and a script in the directory /etc/cron.daily/. This configuration will allow the virtual servers to set the software clock accurately once a day, yet not have the overhead of the chatty Network Time Protocol (NTP).

7.5.11 Setting system to halt on SIGNAL SHUTDOWN

By default, RHEL 5 reboots when a Ctrl-Alt-Del key sequence is trapped. This key sequence is simulated by z/VM when it issues a **SIGNAL SHUTDOWN** command. Rather than rebooting, you want your system to halt (shutdown).

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

```
# cd /etc
# vi inittab    // change shutdown -r to shutdown -h
...
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -h now
...
```

This change will be picked up when the system is rebooted.

7.5.12 Modifying modprobe.conf

When a RHEL 5 system boots it determines which disks to bring online by the device addresses built into the initial RAMdisk. Change the DASD range and rebuild the RAMdisk to pick up that change.

First, edit the dasd= parameter in /etc/modprob.conf. When specifying it you must maintain the ordering of the existing minidisks (/dev/dasda = 100, /dev/dasdb = 101 and /dev/dasdc = 102). Make a backup copy of the file and add the string **dasd=100-102** to the parameter line:

```
# cd /etc
# cp modprobe.conf modprobe.conf.orig
# vi modprobe.conf   // modify the dasd= string
alias eth0 qeth
options dasd_mod dasd=100-102
```

Next, backup the initial RAMdisk and rebuild with **mkinitrd**. This will read modprobe.conf and incorporate the new DASD range.

```
# cd /boot
# cp initrd-2.6.18-1.2839.el5.img initrd-2.6.18-1.2839.el5.img.orig
# mkinitrd -f initrd-2.6.18-1.2839.el5.img $(uname -r)
```

Finally, run the **zipl** command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux' (default)
Preparing boot device: dasda (0100).
Done.
```

7.5.13 Rebooting the system

Now **reboot** to test your changes:

```
# reboot
Broadcast message from root (pts/0) (Sun Nov 19 08:57:32 2006):

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

7.5.14 Verifying the changes

You are now done customizing the master Linux image. When the system comes back up you should verify the changes that you made.

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

   ```
   # df -h
   Filesystem            Size  Used Avail Use% Mounted on
   /dev/dasda1           2.3G  900M  1.3G  42% /
   udev                  247M   72K  247M   1% /dev
   ```

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

   ```
   # swapon -s
   Filename                                Type            Size    Used    Priority
   /dev/dasdb1                             partition       259956  0       -1
   /dev/dasdc1                             partition       519924  0       -2
   ```

   Device with higher priority will be used first, if full, device next in priority is used. If using defaults, the first VDISK (101, **/dev/dasdb1**) gets priority -1 and second VDISK (102, **/dev/dasdc1**) with priority -2 is used after the first is full. You can also specify **pri=nn** in /etc/fstab for swap devices.

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

   ```
   # shutdown -h now
   ```

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

7.5.15 Red Hat bug 224257 workaround

There is a significant bug in the RHEL 5 installer whereby the root file system does not get properly identified as `Linux native`. For details, see the Red Hat Bugzilla bug 224257:

```
https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=224257
```

Use the **fdasd** command against the root file system. You will get a warning that you can heavily damager your system. This is true, but modifying just the partition type will not cause damage. The partition needs to be set to Linux native for installation of the controller to be successful:

```
# fdasd /dev/dasda

WARNING: Your DASD '/dev/dasda' is in use.
         If you proceed, you can heavily damage your system.
         If possible exit all applications using this disk
         and/or unmount it.

reading volume label ..: VOL1
reading vtoc ..........: ok
```

Use the **p** subcommand to print the partition table of `/dev/dasda`:

```
Command action
   m   print this menu
   p   print the partition table
   n   add a new partition
   d   delete a partition
   v   change volume serial
   t   change partition type
   r   re-create VTOC and delete all partitions
   u   re-create VTOC re-using existing partition sizes
   s   show mapping (partition number - data set name)
   q   quit without saving changes
   w   write table to disk and exit

Command (m for help): p

Disk /dev/dasda:
  cylinders ............: 3338
  tracks per cylinder ..: 15
  blocks per track .....: 12
  bytes per block ......: 4096
  volume label .........: VOL1
  volume serial ........: 0X0100
  max partitions .......: 3

 ------------------------------- tracks -------------------------------
              Device      start      end   length   Id  System
          /dev/dasda1         2    50069    50068    1  unknown
```

If the bug exists in the installer, `/dev/dasda1` will be of type `unknown`. If you see a type of `Linux native`, you can type **q** to quit and proceed with the next section.

Use the **t** subcommand to change the file system type of `/dev/dasda1`:

```
Command (m for help): t

Disk /dev/dasda:
  cylinders ............: 3338
  tracks per cylinder ..: 15
  blocks per track .....: 12
  bytes per block ......: 4096
  volume label .........: VOL1
  volume serial ........: 0X0100
  max partitions .......: 3

 ------------------------------- tracks -------------------------------
               Device      start      end   length   Id  System
          /dev/dasda1          2    50069    50068    1  unknown
```

Type **1** to choose the only partition, then type **1** again to set the type to `Linux native`:

```
change partition type
partition id (use 0 to exit): 1

current partition type is: unknown

   1  Linux native
   2  Linux swap
   3  Linux raid

new partition type: 1
```

Type w to write the modified partition table. This will not damage the master image file system, but will allow it to be properly mounted over `/rhel5master/` when installing the controller:

```
Command (m for help): w
writing VTOC...
rereading partition table...

fdasd error:  IOCTL error
Error while rereading partition table.
Please reboot!
```

You can now shutdown the master image:

```
# shutdown -h now
...
```

The partition type of `/dev/dasda1` should now be correct.

7.6 Installing the controller

In this section you will install the controller Linux image.

1. **Logon to RHEL5** or resume your 3270 session. You should see messages as a result of the master image being shut down.

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

2. At this point you could IPL CMS to run the **`RHEL5 EXEC`** again. To review, this EXEC purges the reader, punches the three bootstrap files and IPLs from the reader. Since the same three files should be in your reader, you can save some time by simply IPLing from the reader at virtual address `00C`. Think of it as booting your PC from a floppy disk:

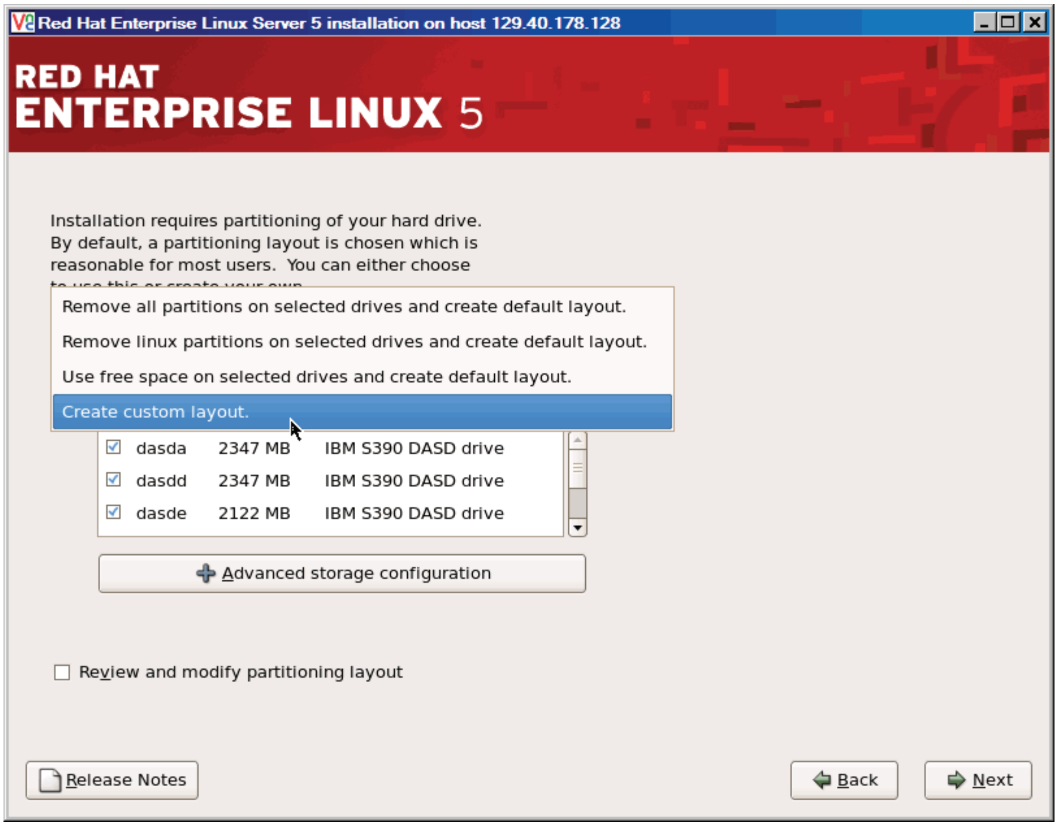
```
==> #cp def stor 512m
==> #cp ipl 00c
00: 0000003 FILES CHANGED

Linux version 2.6.18-8.el5 (brewbuilder@spark.z900.redhat.com) (gcc version 4.1.
1 20070105 (Red Hat 4.1.1-52)) #1 SMP Fri Jan 26 14:16:44 EST 2007
We are running under VM (64 bit mode)
...
```

3. **Install Linux again** as described in “Installing the master image” on page 95. However, more disks will be used. The root file system of the master image is `/dev/dasda1` (minidisk 100). For the controller install, you will set the master image to be mounted as a file system over the directory `/rhel5master/`. In addition to a root file system, a separate minidisk for `/backup/` and a logical volume for `/nfs/` are recommended. The following table is a summary:

Table 7-2 Controller disk partitioning

minidisk	Linux device	Format	Mount Point	Notes
100	`/dev/dasda1`	**No!!**	`/rhel5master`	master image - **don’t format!**
101	`/dev/dasdb1`	N/A	swap	Swap space - set after install
102	`/dev/dasdc1`	N/A	swap	Swap space - set after install
200	`/dev/dasdd1`	As ext3	/	Controller root file system
203	`/dev/dasde1`	As ext3	`/backup`	For backup of z/VM and Linux files
204	`/dev/dasdf1`		`/nfs`	Physical volume 1/4 for logical volume
205	`/dev/dasdg1`		`/nfs`	Physical volume 2/4 for logical volume
206	`/dev/dasdh1`		`/nfs`	Physical volume 3/4 for logical volume
207	`/dev/dasdi1`		`/nfs`	Physical volume 4/4 for logical volume

4. You should see the window that allows you to partition disks as shown in Figure 7-12 on page 118.


Red Hat Enterprise Linux Server 5 installation on host 129.40.178.128
RED HAT
ENTERPRISE LINUX 5
Remove all partitions on selected drives and create default layout.
Remove linux partitions on selected drives and create default layout.
Use free space on selected drives and create default layout.
Create custom layout.
Advanced storage configuration
Release Notes
Back
Next

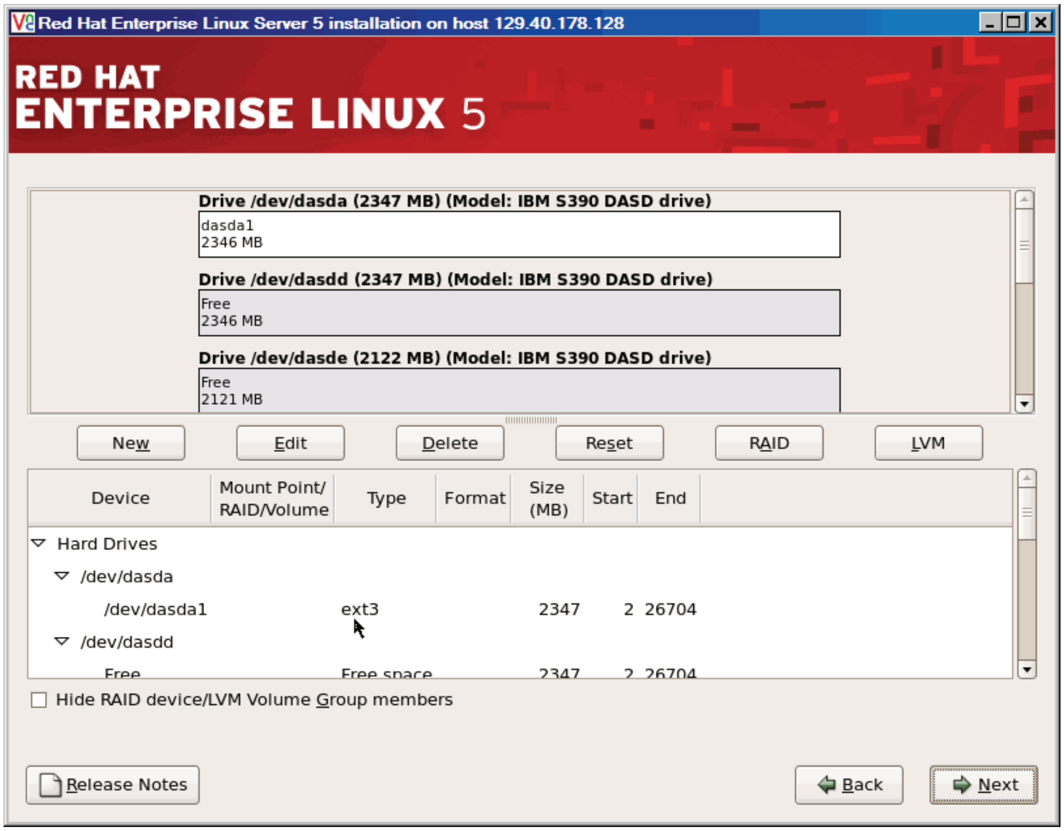
Installation requires partitioning of your hard drive. By default, a partitioning layout is chosen which is reasonable for most users. You can either choose to use this or create your own.

☑	dasda	2347 MB	IBM S390 DASD drive
☑	dasdd	2347 MB	IBM S390 DASD drive
☑	dasde	2122 MB	IBM S390 DASD drive

☐ Review and modify partitioning layout

Figure 7-12 Choosing custom layout

5. select **Choose Custom Layout** from the dropdown menu, then click **Next** .
6. The next window should show your disks as shown in Figure 7-13 on page 119.



Drive /dev/dasda (2347 MB) (Model: IBM S390 DASD drive)
dasda1
2346 MB

Drive /dev/dasdd (2347 MB) (Model: IBM S390 DASD drive)
Free
2346 MB

Drive /dev/dasde (2122 MB) (Model: IBM S390 DASD drive)
Free
2121 MB

Device	Mount Point/ RAID/Volume	Type	Format	Size (MB)	Start	End
Hard Drives						
/dev/dasda						
/dev/dasda1		ext3		2347	2	26704
/dev/dasdd						
Free		Free space		2347	2	26704

□ Hide RAID device/LVM Volume Group members

Figure 7-13 Disk paritioning window

7. You should see `/dev/dasda1` as an ext3 file system (Important: if this shows that /dev/dasda is a Physical Volume (PV), this is due to a bug in the installer. You must end this install and set the partition type of `/dev/dasda1` to be Linux Native as described in section 7.5.15, “Red Hat bug 224257 workaround” on page 115).

8. Double-click `/dev/dasda1`, set the *Mount Point* to **/rhel5master**, accept the default of Leave Unchanged (preserve data) and click **OK**. This will mount the master image’s root file system.

9. Double-click **Free** under `/dev/dasdd`, set the *Mount Point* to **/**, accept the default *File Sytem Type* of ext3 and click **OK**. This will be the root file system of the controller.

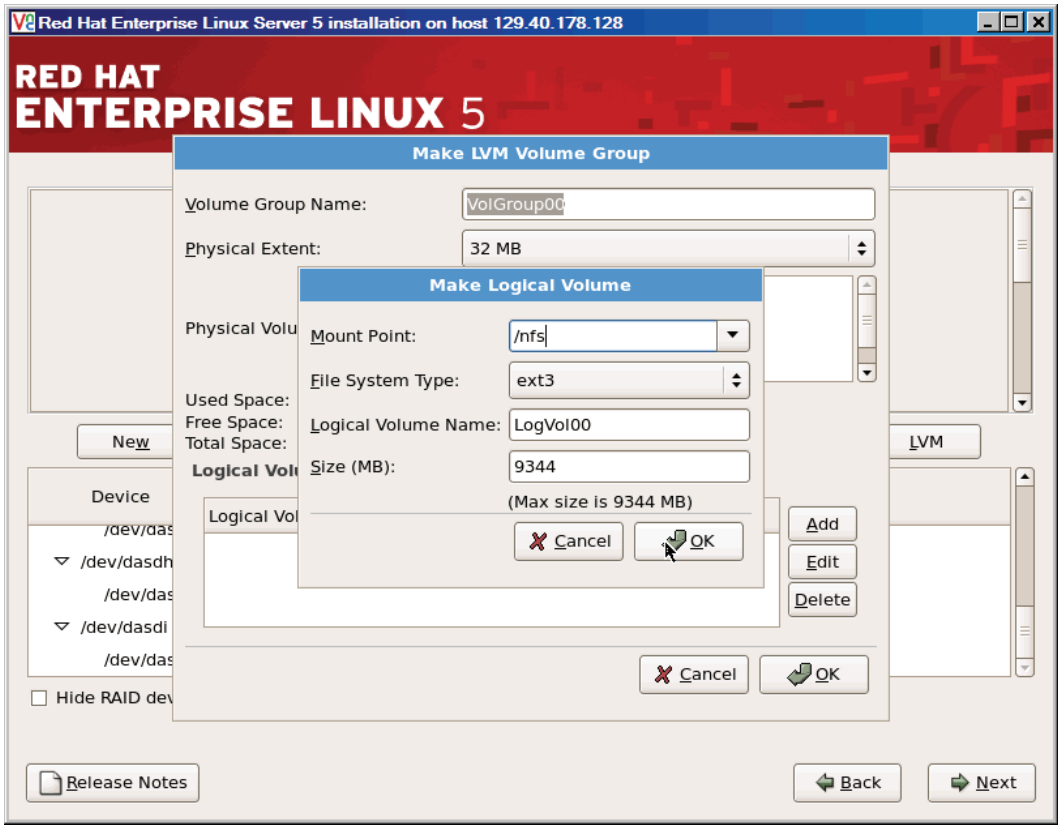
10. Double-click **Free** under `/dev/dasde`, set the *Mount Point* to **/backup**, accept the default *File Sytem Type* of ext3 and click **OK**.

11. Double-cllick **Free** under `/dev/dasdf`, set the *File Sytem Type* to **Physical Volume (LVM)** and click **OK**.

12. Repeat the previous step for `/dev/dasdg`, `/dev/dasdh` and `/dev/dasdi`. This will give you four volumes to create a logical volume.

13. Click the **LVM** button in the upper right. The *Make LVM Volume Group* window should appear.

14. Accept the default *Volume Group Name* of `VolGroup00`. Click the **Add** button on the right. The *Make Logical Volume* window should appear as shown in Figure 7-13 on page 119.



Volume Group Name: VolGroup00

Physical Extent: 32 MB

Mount Point: /nfs

File System Type: ext3

Logical Volume Name: LogVol00

Size (MB): 9344

(Max size is 9344 MB)

Figure 7-14 Configuring a logical volume

15. Back in the *Make LVM Volume Group* window, click **OK**.
16. Back in the partitioning window, click **Next**. You will be warned about not having a swap space. Click **Yes**.
17. In the window that follows, set the fully qualified DNS host name and click **Next**.
18. Set the time zone information and click **Next**.
19. Set the root password and click **Next**.
20. In the Software window, click the **Customize Now** radio button and click Next.
21. To install a minimal image, choose only **Base** and **X Window System** as you did on the controller, and click **Next**.
22. Click **Next** to begin the installation. The installation will continue for a number of minutes. When it is complete, click **Next** and the VNC viewer session will end.
23. Back on you 3270 session, reboot (reIPL) minidisk 200 (the controller), not 100 (the master).

After the system has booted, **login as root** and add the **ARP=no** parameter to the eth0 config file as follows:

```
# echo ARP=no >> /etc/sysconfig/network-scripts/ifcfg-eth0
```

Bring the interface down with the **ifdown** command and back up with the **ifup** command:

```
# ifdown eth0
# ifup eth0
```

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

7.6.1 Verifying the installation

Verify some settings via the **mount** and **df -h** commands. You should see output similar to the following:

```
# mount
/dev/dasdd1 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/dasdg1 on /backup type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
/dev/mapper/VolGroup00-LogVol00 on /nfs type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/dasda1 on /rhel5master type ext3 (rw)
```

This shows file systems mounted over `/`, `/backup/`, `/nfs/` and `/rhel5master/`.

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdd1           2.3G  1.1G  1.1G  49% /
/dev/dasde1           2.1G   68M  1.9G   4% /backup
tmpfs                 250M     0  250M   0% /dev/shm
/dev/mapper/VolGroup00-LogVol00
                      8.9G  149M  8.3G   2% /nfs
/dev/dasda1           2.3G  1.1G  1.1G  50% /rhel5master
```

This shows that the root file system is about half full.

7.7 Configuring the controller

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

- “Copying files to the controller” on page 122
- “Retire the PC NFS server” on page 123
- “Configuring swap spaces” on page 123
- “Configuring yum” on page 123
- “Adding additional RPMs” on page 123
- “Turning off unneeded services” on page 124
- “Configuring the VNC server” on page 125
- “Turning on the NFS server” on page 125
- “Turning on the NTP server” on page 125
- “Setting system to halt on SIGNAL SHUTDOWN” on page 126
- “Configuring SSH keys” on page 126
- “Inserting the vmcp module” on page 127
- “Configuring Apache for DAZL” on page 127
- “Rebooting the system” on page 130

7.7.1 Copying files to the controller

It is now time to copy the RHEL 5 install tree to the controller, along with other files associated with this book. First mount the directory `/nfs/rhel5/` on the PC NFS server over the directory `/mnt/`. In this example the PC NFS server is at IP address 129.40.46.206:

```
# mount <129.40.46.206>:/nfs/rhel5 /mnt
# ls /mnt
EULA             README-or.html          RELEASE-NOTES-ja.html
...
```

Now create a local directory of the same name and recursively copy the tree via the **`cp -a`** command:

```
# mkdir /nfs/rhel5
# cd /mnt
# cp -a * /nfs/rhel5
```

This command will take some time, perhaps 5-10 minutes depending on network speeds.

Now unmount the RHEL 5 install tree and repeat the process to copy the files associated with this book:

```
# cd /
# umount /mnt
# mount <129.40.46.206>:/nfs/virt-cookbook-RH5 /mnt
# mkdir /nfs/virt-cookbook-RH5
# cd /mnt
# cp -a * /nfs/virt-cookbook-RH5
```

Now that the files are copied, you can unmount the `/mnt/` directory. Then view the files that you copied:

```
# cd /nfs/virt-cookbook-RH5
# umount /mnt/
# ls
README.txt   sitar-1.0.7-270.1.noarch.rpm         vm/
rsyncd.conf  virtcook-controller-1-RH5.s390x.rpm
```

The RPM `virtcook-controller-1-rhel5.s390x.rpm`, contains files for use on the controller. The controller RPM contains the following:

```
# rpm -qlp virtcook-controller-1-RH5.s390x.rpm
/backup/linux
/backup/vm
/etc/cron.daily/nightly.sh
/sbin/backupLinux.sh
/sbin/backupVM.sh
/sbin/clone.sh
/sbin/getVMinfo.sh
/var/www/cgi-bin/catSitar.sh
/var/www/cgi-bin/dazd.sh
/var/www/cgi-bin/dazl.sh
...
/var/www/html/editLog.js
/var/www/html/index.html
/var/www/html/xhr.js
```

The first two entries are empty directories created for backup purposes. Each of the remaining files will be addressed in the sections that follow. From the controller, install the RPM with the following command:

```
# rpm -Uvh virtcook-controller-1-RH5.s390x.rpm
Preparing...                ########################################### [100%]
   1:virtcook-controller    ########################################### [100%]
```

7.7.2 Retire the PC NFS server

You have now copied all files related to this book to the controller and you have set the master image automount configuration file to point to it. You should be in a position to retire your PC NFS server, if you desire.

7.7.3 Configuring swap spaces

Turn on the two swap spaces just as you did in section 7.5.1, “Configuring VDISK swap” on page 106.

7.7.4 Configuring yum

You will again configure **yum** so it can install RPMs from automounted install tree. Note that the configuration is identical to the master image because in both instances the install tree is in the directory /nfs/rhel5/. However on the controller this directory is local, while on the master image (and later the clones) the directory is automounted.

Create a file named rhel5.repo in the /etc/yum.repos.d directory:

```
# cd /etc/yum.repos.d
# vi rhel5.repo
[RHEL5]
name=Red Hat Enterprise Linux 5
baseurl=file:///nfs/rhel5/Server
```

Import the RPM key:

```
# cd /nfs/rhel5
# rpm --import RPM-GPG-KEY
```

Note: For a RHEL 5 beta distribution, use the command **rpm --import RPM-GPG-KEY-redhat-beta**

7.7.5 Adding additional RPMs

It is recommended that you install the same packages as you did on the master image. Refer back to “Configuring automount of install tree” on page 107 and “Adding additional RPMs” on page 108. Since you just copied the install tree from the PC NFS server to the controller, you can point **yum** to the new /nfs/rhel5/ directory.

Use the **yum -y install** command to install the openmotif and xinetd packages from the local RPM repository and the **rpm -Uvh** command to install the sitar package:

```
# yum -y install openmotif xinetd
...
# cd /nfs/virt-cookbook-RH5
# rpm -Uvh sitar-1.0.7-270.1.noarch.rpm
Preparing...                ########################################### [100%]
   1:sitar                  ########################################### [100%]
```

Important: if you get an error of the following form:

```
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for openmotif to pack into transaction set.
media://1169877309.407096%233/openmotif-2.3.0-0.3.el5.s390x.rpm: [Errno 4] IOError:
<urlopen error unknown url type: media>
Trying other mirror.
Error: failed to retrieve openmotif-2.3.0-0.3.el5.s390x.rpm from RHEL5
error was [Errno 4] IOError: <urlopen error unknown url type: media>
```

There is a problem. The repository can be built to workaround this problem. First the `createrepo` RPM must be installed:

```
# cd /nfs/rhel5/Server/
# rpm -Uvh createrepo-0.4.4-2.fc6.noarch.rpm
warning: createrepo-0.4.4-2.fc6.noarch.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
Preparing...                ########################################### [100%]
   1:createrepo             ########################################### [100%]
```

Now, rename the `repodata/` directory and create a new one via the **`createrepo`** command:

```
# mv repodata/ repodata.orig
# createrepo .
844/2495 -
jakarta-commons-collections-testframework-javadoc-3.1-6jpp.1.s390x.rp2495/2495 -
rsh-server-0.17-37.el5.s390x.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

The new repository should work for any **`yum`** operations, but if you ever want to reinstall with this tree, replace the `repodata/` with `repodata.orig/`.

7.7.6 Turning off unneeded services

As with the master image, follow the steps in “Turning off unneeded services” on page 109. Following is a summary:

```
# chkconfig cups off
# chkconfig iptables off
# chkconfig ip6tables off
# chkconfig auditd off
# chkconfig haldaemon off
# chkconfig atd off
# chkconfig kudzu off
# chkconfig mdmonitor off
# chkconfig rpcgssd off
# chkconfig rpcidmapd off
# chkconfig anacron off
# chkconfig mcstrans off
# chkconfig yum-updatesd off
```

7.7.7 Configuring the VNC server

Follow the same steps as described in 7.5.7, “Configuring the VNC server” on page 110, with a small difference, you don’t have to remove **/root/.vnc/passwd** file at the end.

7.7.8 Turning on the NFS server

The NFS server will be needed to export the RHEL 5 install tree and the files associated with this book to the other virtual servers. Enable NFS as follows:

```
# vi /etc/exports
/nfs/rhel5            *(ro,sync)
/nfs/virt-cookbook-RH5 *(ro,sync)
```

These two lines will cause NFS to export:

- The /nfs/rhel5/ directory, which contains the Red Hat Enterprise Linux 5 installation.
- The /nfs/virt-cookbook-RH5/ directory, which has the files associated with this book.

Set the NFS server to start at boot time and for this session.

```
# chkconfig nfs on
# service nfs start
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS daemon: [ OK ]
Starting NFS mountd: [ OK ]
```

You could test mounting the directories locally if you choose to.

7.7.9 Turning on the NTP server

It is desirable to have all the system clocks on your Linux images be the same. This can be accomplished by setting up a time server on the controller, and have the other virtual servers synchronize their times against the controller. The controller’s time can be set accurately by having it synchronize against public time servers on the Internet. If you do want to access an external public time server, then make sure your firewall allows traffic through between the controller and the public time servers on the well-known NTP port, 123.

There are basicaly two possibilities how to connect to public NTP servers on Internet. Either chosing random servers from public list (your system will be then synchronized against only those servers) or using pool of NTP servers Using the pool of servers is recommended to avoid the so called *public server abuse*. You can read more on this at:

http://www.pool.ntp.org/

Test to see that the pool of servers is indeed accessible by vai the **ntpdate** comnand:

```
# ntpdate pool.ntp.org
22 Dec 10:46:12 ntpdate[32170]: step time server 209.223.236.234 offset -60.787960 sec
```

Start the **ntpd** server on the controller at boot time via the **chkconfig** command and for this session via the **service** command:

```
# chkconfig ntpd on
# service ntpd start
Starting ntpd: [  OK  ]
```

If you immediately run **ntptrace**, you might see that your time server is stratum 16 which means that the clock is not set accurately:

```
# ntptrace
```

```
localhost: stratum 16, offset 0.000000, synch distance 0.001170
```

After you start **ntpd**, you will have to wait some time for the clock to be adjusted. Use the **ntpq** command with the **peers** subcommand to show the two NTP servers that were set in the /etc/ntp.conf file:

```
# ntpq
ntpq> peers
remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 alien.astrosmo. 38.14.160.172    2 u   61   64    3  124.862    2.762   3.920
 163.25.109.18   140.112.4.189    2 u   61   64    3  301.322  -51.900   3.696
 uicsgtw.cs.ui.a .INIT.          16 u    -   64    0    0.000    0.000 4000.00
ntpq> quit
```

You will see different IP addresses because pool of servers is random. After a few minutes (can take up to half an hour), the **ntptrace** command should show that your system is now a stratum 2 server:

```
# ntptrace
localhost: stratum 2, offset -0.002555, synch distance 0.006114
...
```

7.7.10 Setting system to halt on SIGNAL SHUTDOWN

Again, RHEL5 reboots when a Ctrl-Alt-Del key sequence is trapped. This key sequence is simulated by z/VM when a **SIGNAL SHUTDOWN** command is issued. Rather than rebooting, you want your system to halt (shutdown). Change this setting by changing **shutdown -r** to **shutdown -h** in the /etc/inittab file:

```
# cd /etc
# vi inittab    // change shutdown -r to shutdown -h
...
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -h now
...
```

This change will be picked up when the system is rebooted.

7.7.11 Configuring SSH keys

SSH sessions are typically authenticated via passwords typed in from the keyboard. With SSH *key-based authentication* sessions can be authenticated via public and private keys so that no password is needed. To accomplish this, the following must be true:

- The SSH server must have the client's public key.
- The SSH client must send its private key.
- The keys must match.

SSH key-based authentication can be set up from the controller (client) to the virtual servers. If the master image has a copy of controller's public key in the file /etc/ssh/authorized_keys, and the controller has a symbolic link to its private key in the file /root/.ssh/id_dsa, then key based authentication will work to the cloned virtual servers.

Create a new DSA key in the directory /etc/ssh/:

```
# cd /etc/ssh
# ssh-keygen -t dsa -P "" -f id_dsa
Generating public/private dsa key pair.
Your identification has been saved in id_dsa.
Your public key has been saved in id_dsa.pub.
The key fingerprint is:
c1:98:6e:c1:48:52:80:56:98:be:67:aa:1d:d5:76:95 root@lat127.pbm.ihost.com
```

This creates a key pair where the file with the `.pub` suffix is the public key and the other file is the private key. Note that the private key is only readable by root:

```
# ls -l id_dsa*
-rw------- 1 root root 668 Dec 22 13:47 id_dsa
-rw-r--r-- 1 root root 615 Dec 22 13:47 id_dsa.pub
```

Create a new directory on the master image where the public key will be copied:

```
# mkdir /rhel5master/root/.ssh
```

Copy the public key with the name `authorized_keys`:

```
# cp id_dsa.pub /rhel5master/root/.ssh/authorized_keys
```

Make a symobolic link to the private key, `id_dsa`, in the new directory `/root/.ssh/`:

```
# mkdir /root/.ssh
# cd /root/.ssh
# ln -s /etc/ssh/id_dsa
```

7.7.12 Inserting the vmcp module

To issue CP commands the vmcp module is needed. By default it is not loaded at boot time. One way to accomplish this is to add the **`modprobe vmcp`** command, which will insert the module, to the file `/etc/rc.d/rc.local` which is run at boot time:

```
# cd /etc/rc.d
# vi rc.local   // add one line
...
touch /var/lock/subsys/local
modprobe vmcp
```

The **`vmcp`** command will now be available after the next reboot.

7.7.13 Configuring Apache for DAZL

A working Apache Web server is needed for the Web application named *Data About z/VM and Linux (DAZL)*. For details on this application, see section 13.5, “Monitoring z/VM and Linux via a Web application” on page 211. Because this application exposes sensitive data, the entire Web site should be secured. To accomplish this, a valid user ID/password pair should be required to get to the Web site.

Installing Apache

First, install the Apache Web server RPMs via the **`yum -y install`** command:

```
# yum -y install httpd httpd-manual
...
Running Transaction
  Installing: apr                          ######################### [1/5]
  Installing: postgresql-libs              ######################### [2/5]
  Installing: apr-util                     ######################### [3/5]
  Installing: httpd                        ######################### [4/5]
  Installing: httpd-manual                 ######################### [5/5]

Installed: httpd.s390x 0:2.2.3-6.el5 httpd-manual.s390x 0:2.2.3-6.el5
```

```
Dependency Installed: apr.s390x 0:1.2.7-10 apr-util.s390x 0:1.2.7-3
postgresql-libs.s390x 0:8.1.4-1.1
Complete!
```

Securing the Web site

Now that the Apache RPMs are installed, it is recommended that you secure the Web site because the DAZL application exposes much sensitive data. Create a password file that Apache will use to secure the Web site. The **htpasswd** command is used to create a password file named `.htpasswd` in Apache's default DocumentRoot directory, `/var/www/htdocs/`.

Change to the document root directory where you should see three files that were installed with the controller RPM. These are part of the DAZL application:

```
# cd /var/www/html
# ls
editLog.js  index.html  xhr.js
```

Create a password file for root via the **htpasswd** command:

```
# htpasswd -c .htpasswd root
New password:
Re-type new password:
Adding password for user root
```

Once the password file is created, the configuration file `/etc/httpd/conf/httpd.conf` is modified so the DocumentRoot `Directory` statement will require valid credentials in this password file. Make a backup copy of the file then add four lines and optionally one comment:

```
# cd /etc/httpd/conf
# cp httpd.conf httpd.conf.orig
# vi httpd.conf   // Add 4 lines and one comment to require correct user ID and password
... // around line 300
<Directory "/var/www/html">
...
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
    AllowOverride None

# add 4 - make this a restricted Web site
    AuthType Basic
    AuthName "This is a restricted Web site"
    AuthUserFile /var/www/html/.htpasswd
    Require user root
#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all

</Directory>
...
```

Allowing Apache to run CP and SSH commands

The DAZL application uses CP commands via the **vmcp** command to get z/VM data, and the **ssh** command with key-based authentication to get Linux data. By default the **httpd** service runs as the user `apache` which does not have the permission to run these commands. In order to authorize the Web server to do this, two changes must be made

- The **sudo** command is used to allow access to **vmcp** and **ssh**
- SELinux is set to permissive mode

Attention: It is possible to configure SELinux to allow sudo access while still in enforcing mode, but somewhat complicated. If you want to leave SELinux in enforcing mode, see the following related Web page on how to allow **sudo** commands:

```
http://kbase.redhat.com/faq/FAQ_49_6086.shtm
```

Modify the /etc/sudoers configuration file to allow this access. This is done via the **visudo** command so the configuration file is locked. The line with the **requiretty** directive (around line 56) must be commented as the Web server cannot run from a TTY. Then two lines must be added at the bottom of the file to allow the apache uid to run **vmcp** and **ssh**:

```
# cd /etc
# cp sudoers sudoers.orig
# visudo   // Comment “requiretty” near line 56 and add three lines at the bottom
...
# Defaults    requiretty
...
## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

# allow www group to run vmcp and ssh so Apache2 can populate DAZL pages
apache     ALL=NOPASSWD:/sbin/vmcp
apache     ALL=NOPASSWD:/usr/bin/ssh
```

Make a backup copy of the file /etc/selinux/config and change the SELINUX variable from enforcing to permissive:

```
# cd /etc/selinux
# cp config config.orig
# vi config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=permissive
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Now Apache should be able to run **vmcp** and **ssh** commands via **sudo**.

Setting ownership of Linux backup directories

In order for DAZL to be able to update the description/log files of each virtual server, the Apache Web server will need permission to write to those files. Apache runs as the group named apache so one way to allow this is via the group write and setuid bits. Use the following **chgrp** and **chmod** commands to accomplish this:

```
# cd /backup
# chgrp apache linux
# chmod g+ws linux
```

The **chgrp** command sets the group owner of the directory /backup/linux/ to apache. Setting the group write (g) bit gives that group write permissions and setting the group setuid (s) bit pushes that permission down to newly created directories.

Starting Apache

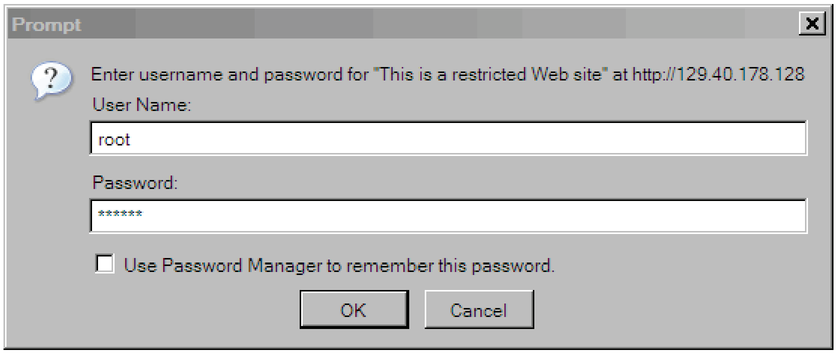
First stop the firewall if it is running

```
# service iptables stop
```

Start Apache for this session with the **httpd** service script. Also, use the **chkconfig** command so that the Apache Web server starts when the system is booted:

```
# service httpd start
Starting httpd: [  OK  ]
# chkconfig httpd on
```

Now the Apache Web server will start and will require the specified user ID/password pair for anyone to get access. Point a browser to the IP address of the controller. You should be prompted for credentials. The following figure shows the prompt window from a FireFox browser:



Enter username and password for "This is a restricted Web site" at http://129.40.178.128

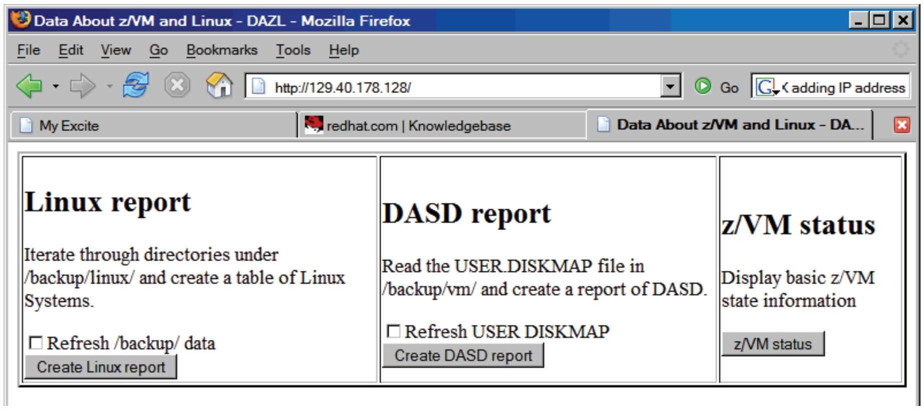
User Name: root

Password: ******

☐ Use Password Manager to remember this password.

Figure 7-15   Browser prompting for credentials

When you supply the correct credentials, you should see the DAZL home page as shown in Figure 7-16:



Linux report

Iterate through directories under /backup/linux/ and create a table of Linux Systems.

☐ Refresh /backup/ data

DASD report

Read the USER.DISKMAP file in /backup/vm/ and create a report of DASD.

☐ Refresh USER DISKMAP

z/VM status

Display basic z/VM state information

Figure 7-16   DAZL home page

You should now have Apache configured for DAZL. Before you create any clones, there is not a lot that this application will do for you. You can utilize this application later in the book in section 13.5, “Monitoring z/VM and Linux via a Web application” on page 211.

7.7.14 Rebooting the system

You should now reboot the system to test the changes:

```
# reboot
```

After your system comes back in a couple of minutes, start a new SSH session to the controller.

7.7.15 Verifying the changes

You are now done customizing the controller Linux image. **SSH back into the controller** and check a few settings. Test the **vmcp** command with a CP command such as **QUERY NAMES**:

```
# vmcp q n
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
RHEL5  - DSC
VSM     - TCPIP
```

Confirm that both of your swap spaces are operational:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdb1                             partition       259956  0       -1
/dev/dasdc1                             partition       519924  0       -2
```

Verify the NFS server is running:

```
# service nfs status
rpc.mountd (pid 6776) is running...
nfsd (pid 6770 6769 6768 6767 6766 6765 6764 6763) is running...
rpc.rquotad (pid 6748) is running...
```

Congratulations! You have installed and configured Linux twice onto the RHEL5 user ID. You are now ready to clone the master image to a new virtual server.

8

Configuring RHEL 5 for cloning

It has become appallingly obvious that our technology has exceeded our humanity.

--Albert Einstein

The RHEL5 user ID is now customized with both a master image and a controller. The controller should now be running. In this chapter, you will perform the following steps:

- "Defining a new user ID for a virtual server" on page 133
- "Cloning a virtual server manually" on page 136
- "Cloning a virtual server automatically" on page 140
- "Cloning five more virtual servers" on page 142

8.1 Defining a new user ID for a virtual server

In this section you will define a new user ID, LINUX01, in z/VM and clone the master image to it.

1. Logon to MAINT and edit the USER DIRECT file to add more Linux ID's.

   ```
   ==> x user direct c
   ```

2. Go to the bottom of the file and add the following 5 lines. In this example the user ID will be LINUX01 with a password of LNX4VM. It will default to have 256MB of memory but can be set up to 1GB. It will have only G permission (General user) It will have a 3338 cylinder (about 2.2 GB) minidisk for the root file system. In this example it is at device address A740 which was formatted and given a label of MMA740 earlier:

   ```
   USER LINUX01 LNX4VM 256M 1G G
    INCLUDE LNXDFLT
    OPTION APPLMON
    MDISK 100 3390 0001 3338 <MMA740> MR LNX4VM LNX4VM LNX4VM
   ```

3. Add the new volume to the $ALLOC$ user ID so cylinder 0 won't show up in the disk map as a gap. Save your changes with the **FILE** subcommand:

   ```
   ====> top
   ====> /alloc
   USER $ALLOC$  NOLOG
   MDISK A01 3390 000 001 MVA700 R
   ```

```
...
 MDISK A0B 3390 000 001 <MMA740> R
====> file
```

4. Again check for gaps and overlaps. You can use the **ALL** subcommand with the logical OR operator "|" to check for both strings. You should see only one 501 cylinder gap.

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

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

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

The new Linux user ID has now been defined.

8.1.1 Adding LINUX01 to AUTOLOG1's PROFILE EXEC

The new Linux ID you defined needs access to the VSWITCH. A **SET VSWITCH** command with the **GRANT** parameter can be added to AUTOLOG1's **PROFILE EXEC** to do this. Also, an **XAUTOLOG** statement can be added if the user ID is automatically logged on at z/VM IPL time:

Link and access the AUTOLOG1 191 disk read/write and edit the file **PROFILE EXEC**. Add LINUX01 to the sections that grant access to the VSWITCH and that **XAUTOLOG** the Linux user IDs:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
==> x profile exec f   // add two lines
/***************************/
/*  Autolog1 Profile Exec  */
/***************************/
'cp xautolog tcpip'                /* start up TCPIP */
'CP XAUTOLOG DTCVSW1'              /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVSW2'              /* start VSWITCH controller 2 */
'cp set pf12 ret'                  /* set the retrieve key */
'cp set mdc stor 0m 128m'          /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m'          /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
'cp set signal shutdown 180'       /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
'cp set vswitch vsw1 grant rhel5'
'cp set vswitch vsw1 grant linux01'

/* XAUTOLOG each Linux user that should be started */
'cp xautolog rhel5'
'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
```

8.1.2 Creating a parameter file for the new LINUX ID

For each Linux guest you want to clone, you need to create a parameter file. This file specifies many of the installation parameters. It will be used both when cloning to this user ID and when installing RHEL 5 manually.

1. Logon to `LNXMAINT`.
2. Copy an existing configuration file and edit the new file to apply to the new Linux. If the new Linux is going to be on the same network as the controller you will likely only have to change the `HOSTNAME` and `IPADDR` variables:

```
==> copy rhel5 conf-rh5 d linux01 = =
==> x linux01 conf-rh5 d
DASD=100-106
HOSTNAME=lat121.pbm.ihost.com
NETTYPE=qeth
IPADDR=129.40.178.121
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETWORK=129.40.178.0
NETMASK=255.255.255.0
SEARCHDNS=pbm.ihost.com
BROADCAST=129.40.178.255
GATEWAY=129.40.178.254
DNS=129.40.106.1
MTU=1500
PORTNAME=DONTCARE
```

3. Copy an existing parameter file and edit the new file to apply to the new Linux. Modify the file name of the configuration file.

```
==> copy rhel5 parm-rh5 d linux01 = =
==> x linux01 parm-rh5 d
ramdisk_size=40000 root=/dev/ram0 ro ip=off
CMSDASD=191 CMSCONFFILE=LINUX01.CONF-RH5
vnc vncpassword=lnx4vm
```

4. Logoff of `LNXMAINT`.
5. Logon to `LINUX01`.
6. Answer **N** to the question “`Do you want to IPL Linux from DASD 100`”. Verify that the new Linux user ID has a NIC at addresses 600-602:

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
00: built on IBM Virtualization Technology
...
```

7. Verify that the minidisk at addresses `100` and the VDISKs at address `101` and `102` are read/write:

```
==> q da
00: DASD 0100 3390 MMA740 R/W       3338 CYL ON DASD  A740 SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000D
00: DASD 0102 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 000E
00: DASD 0190 3390 MVA700 R/O        107 CYL ON DASD  A700 SUBCHANNEL = 0008
00: DASD 0191 3390 MMA70C R/O        300 CYL ON DASD  A70C SUBCHANNEL = 000B
...
```

8. Logoff `LINUX01`.

You should now be ready to clone to this new user ID.

8.2 Cloning a virtual server manually

Before using the shell script **clone.sh** to clone a server, it is recommended that you 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 Linux image has a single root file system on minidisk 100.
- The target user ID, LINUX01 in this example, has identically sized mindisk 100.
- The **vmcp** command is available to issue z/VM CP commands
- The z/VM **FLASHCOPY** command can be used but if you don't have that support, the Linux **dd** command will work.

Given these assumptions, one set of steps that can be used to clone a system is as follows:

1. Link the target disk read/write.
2. Copy the source to the target disk via **FLASHCOPY** or the Linux **dd** command.
3. Mount the newly copied root file system.
4. Modify the networking information on the target system.
5. Detach the target disk.
6. IPL the target system.
7. Modify the SSH keys on the target system.

Link the target disk read/write

Start an SSH session to the controller as root.

The target root file system, LINUX01 100, is linked multi-read (read/write if no other user ID has write access) as virtual device 1100 via the CP **LINK** command:

```
# vmcp link linux01 100 1100 mr
```

Copy the source to the target disk via FLASHCOPY

The two disks are copied via the CP **FLASHCOPY** command:

```
# vmcp flashcopy 100 0 end to 1100 0 end
Command complete: FLASHCOPY 100 0 END TO 1100 0 END
```

Attention: If you do not have **FLASHCOPY** support, you can use the Linux **dasdfmt** and **dd** commands. You must first enable the 1100 disk via the **chccwdev -e** command, then determine the newly created device node via the **lsdasd** command:

```
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
# lsdasd
0.0.0100(ECKD) at (94:0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at (94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at (94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
...
0.0.1100(ECKD) at (94:36) is dasdj : active at blocksize 4096, 600840 blocks, 2347 MB
```

In this example the source minidisk (100) is named `/dev/dasda`, while the target minidisk (1100) is named `/dev/dasdj`. Format the device via the dasdfmt command using a 4096 byte (4KB) block size:

```
# dasdfmt -b 4096 -y -f /dev/dasdj
Finished formatting the device.
Rereading the partition table... ok
```

Now that the device has been formatted, you can copy the contents of the master image via the **dd** command, again using a block size of 4K (4096) bytes:

```
# dd if=/dev/dasda of=/dev/dasdj bs=4096
600840+0 records in
600840+0 records out
2461040640 bytes (2.5 GB) copied, 81.2098 seconds, 30.3 MB/s
```

Then bring the device offline so the new file system will be recognized when it is brought back online:

```
# chccwdev -d 1100
Setting device 0.0.1100 offline
Done
```

Mount the newly copied root file system

Enable the new minidisk at virtual device address 1100:

```
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
```

Determine which device node is used.

```
# lsdasd
0.0.0100(ECKD) at (94:0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at (94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at (94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0200(ECKD) at (94:12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0203(ECKD) at (94:16) is dasde : active at blocksize 4096, 543240 blocks, 2122 MB
0.0.0204(ECKD) at (94:20) is dasdf : active at blocksize 4096, 840060 blocks, 3281 MB
0.0.0205(ECKD) at (94:24) is dasdg : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0206(ECKD) at (94:28) is dasdh : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0191(ECKD) at (94:32) is dasdi : active at blocksize 4096, 54000 blocks, 210 MB
0.0.1100(ECKD) at (94:36) is dasdj : active at blocksize 4096, 600840 blocks, 2347 MB
```

In this case, the device node is `/dev/dasdj`. Because there is only one partition in the root file system, it will be `/dev/dasdj1`. Mount it over a newly created mount point `/mnt/linux01`

```
# mkdir /mnt/linux01
# mount /dev/dasdj1 /mnt/linux01
```

Observe that this appears to be a root file system:

```
# ls /mnt/linux01/
bin   dev  home  lib64       media  opt   root  selinux  sys  usr
boot  etc  lib   lost+found  mnt    proc  sbin  srv      tmp  var
```

Modify networking information on the target system

In this example, the only two pieces of networking information that are modified are the IP address (from `129.40.178.128` to `129.40.178.121`) and the host name (from `lat128` to `lat121`).

The host name and IP address are changed in the file `/etc/hosts`:

```
# cd /mnt/linux01/etc
# vi hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1              localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
129.40.178.121         lat121.pbm.ihost.com lat121
```

The host name is changed in the file `/etc/sysconfig/network`:

```
# cd sysconfig
# vi network
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=lat121.pbm.ihost.com
GATEWAY=129.40.178.254
```

The IP address is changed in the file `/etc/sysconfig/network-scripts/ifcfg-eth0`:

```
# cd network-scripts
# vi ifcfg-eth0
# IBM QETH
DEVICE=eth0
BOOTPROTO=static
IPADDR=129.40.178.121
MTU=1500
NETMASK=255.255.255.0
NETTYPE=qeth
ONBOOT=yes
PORTNAME=DONTCARE
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
MTU=1500
ARP=no
```

Detach the target disk

Change to the default directory, use the **sync** command to flush the disks and the **umount** command to unmount the modified root file system:

```
# cd
# sync
# umount /mnt/linux01
```

Set the LINUX01 100 disk offline with the **chccwdev** command and detach it via the CP **DETACH** command:

```
# chccwdev -d 1100
Setting device 0.0.1100 offline
Done
# vmcp det 1100
DASD 1100 DETACHED
```

You should now be ready to IPL the manually cloned system.

IPL the target system

Logon to a 3270 session as LINUX01. CMS will IPL and the **PROFILE EXEC** will ask you if you want to IPL from minidisk 100. Type **y** for yes then you can enter the command **#cp vi vmsg 0** to select the default Linux image from the boot menu:

```
LOGON LINUX01
ENTER PASSWORD  (IT WILL NOT APPEAR WHEN TYPED):

00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (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 16:03:14 EST TUESDAY 12/12/06
z/VM V5.2.0    2006-07-18 08:48

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 100? y/n
y
00: zIPL v1.5.3 interactive boot menu
00:
00:  0. default (linux)
00:
00:  1. linux
00:
00: Note: VM users please use '#cp vi vmsg <input>'
00:
00: Please choose (default will boot in 15 seconds):
00:
00: CP VI VMSG 0
00: Booting default (linux)...
Linux version 2.6.18-1.2839.el5 (brewbuilder@devel8.z900.redhat.com) (gcc versio
n 4.1.1 20061130 (Red Hat 4.1.1-43)) #1 SMP Tue Dec 5 22:58:42 EST 2006
We are running under VM (64 bit mode)
Detected 2 CPU's
Boot cpu address  0
...
Red Hat Enterprise Linux Server release 4.92 (Tikanga)
Kernel 2.6.18-1.2839.el5 on an s390x

lat121 login:
```

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 to the new clone as root.

The SSH keys that were copied are identical to those of the master 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 master image except for the IP address and host name.

Next you will 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 log off the LINUX01 user ID:

```
==> #cp log
```

8.3 Cloning a virtual server automatically

Now that you have cloned a server manually and better understand the steps, you can use the **clone.sh** script to clone automatically.

Go back to your an SSH session to the controller. The **clone.sh** script should be in your PATH in the directory /usr/local/sbin/. You can verify this with the **which** command:

```
# which clone.sh
/sbin/clone.sh
```

The script takes one parameter which is the Linux user ID that the master image will be cloned to. That user ID must be logged off. It reads the parameter file on the LNXMAINT 192 disk (the controller's 191 disk) to obtain information necessary to give the new Linux virtual server an identity. It calls CP **FLASHCOPY** via the **vmcp** module/command to try to copy the 100 (master image) disks. If **FLASHCOPY** fails, the script falls back to copying the disk via the Linux **dasdfmt** and **dd** commands. The script then boots the new Linux via the **XAUTOLOG** command. It also creates an empty backup directory under /backup/linux/ and adds the server's public key is added to the controller's known_hosts file.

It should take less than a minute to clone with **FLASHCOPY** support and 3-20 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 linux01
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
LINUX01 CONF-RH5 D1 F        80         13          1 12/12/2006 13:54:35
```

```
WARNING!!: this will copy disk 100 to LINUX01 100
New host name will be:  lat121.pbm.ihost.com
New TCP/IP address will be: 129.40.178.121
Other network data is retrieved from LINUX01 CONF-RH5 on 191 disk
Are you sure you want to overwrite these disks (y/n):
```

In the section of output above, the script makes sure the user ID to be cloned to exists and is logged off. It then searches for the correct PARMFILE and obtains the necessary networking information. It then asks if you are sure you want to overwrite the disks on the target user ID.

```
Please enter a brief description of this server
This is a Web Server running RHEL 5
```

You are then prompted to enter a one line description of the server. This will be saved on the controller in the file /backup/linux/<userid>-on-<tcp.ip.addr>. This file is used by the DAZL Web application:

```
Copying 0100 root file system to LINUX01 ...
Invoking CP command: QUERY VIRTUAL 1100
HCPQVD040E Device 1100 does not exist
Error: non-zero CP response for command 'QUERY VIRTUAL 1100': #40
Invoking CP command: LINK LINUX01 0100 1100 MR

Invoking CP command: FLASHCOPY 0100 0 END 1100 0 END
Command complete: FLASHCOPY 0100 0 END TO 1100 0 END
Invoking CP command: DETACH 1100
DASD 1100 DETACHED
Copying disk via FLASHCOPY succeeded ...
Mounting newly cloned image over /mnt/rhel5cloned ...
Invoking CP command: LINK LINUX01 100 1100 MR
```

In the section above the master image root file system is copied to the LINUX01 100 minidisk via the z/VM **FLASHCOPY** command. Should this fail (for example if you don't have FLASHCOPY support), the code will fall back to the **dasdfmt** and **dd** commands to perform the copies, which will take a number of minutes.

```
Setting device 0.0.1100 online
Done
Mounting /dev/dasdm1 over /mnt/rhel5cloned ...
Modifying cloned image under /mnt/rhel5cloned ...
Regenerating SSH keys in /mnt/rhel5cloned/etc/ssh/ ...
Adding 129.40.178.121 to known_hosts file
Setting device 0.0.1100 offline
Done
Invoking CP command: DETACH 1100
DASD 1100 DETACHED
```

In the section above, the newly cloned file system (LINUX01 100) is linked, activated and mounted over a temporary directory /mnt/rhel5cloned/. Then networking information is modified in the files /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0 and /etc/hosts. Then the SSH keys are regenerated so they are unique for the new virtual server. Then the server's public key is added to the controller's /root/.ssh/known_hosts file. Finally the new disk is set offline and detached.

```
Invoking CP command: XAUTOLOG LINUX01
Command accepted
Successfully cloned /rhel5master to LINUX01
You should be able to ping 129.40.178.121 within one minute
```

In the final section, the target user ID is logged on via **XAUTOLOG**. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. A directory with the user ID and IP address in the name is created under /backup/linux/. The new clone should be on the network in less than a minute.

> **Note:** If the **clone.sh** script fails you can also add the **-v** flag for some more diagnostics. Also, check that:
>
> - The target user ID has been granted access to the VSWITCH
> - The parameter and configuration files are copied and set correctly on LNXMAINT 192

Following is a block diagram of this process both for the manual clone and using the **clone.sh** script:

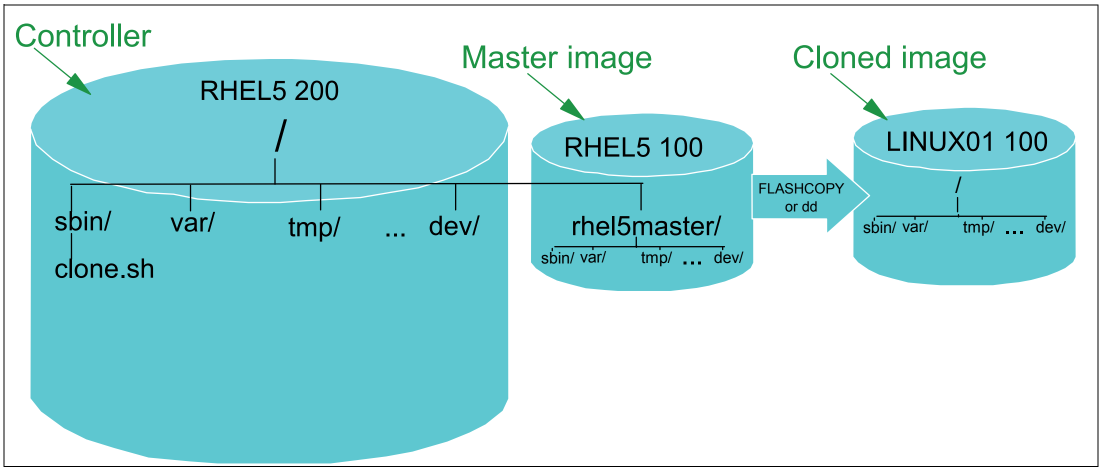


Figure 8-1 Cloning block diagram

The left side of the figure shows the controller which is on the RHEL5 200 disk. It has the master image mounted over the directory /rhel5master/, which is the RHEL5 100 disk. The VDISK swap spaces, RHEL5 101 and 102, are created in memory, so they do not need to be copied.

The script **/sbin/clone.sh** is invoked and it uses either **CP FLASHCOPY** or the Linux **dd** command to copy the 100 minidisk to the target z/VM user ID. The script then mounts the newly copied 100 disk and modifies the networking information to use those values found in the parameter file on the LNXMAINT 192 disk. The script then invokes the **CP XAUTOLOG** command to log that user ID on. Because the user ID is logged on disconnected, the common **PROFILE EXEC** from the LNXMAINT 192 disk IPLs from virtual device address 100 and the newly cloned Linux system is brought to life.

8.4 Cloning five more virtual servers

So far you have installed Linux manually twice on RHEL5 to create a master image and a controller. You have created a new user ID LINUX01 and cloned it. Now it is time to clone five more times. This will give you six virtual servers described in this book. The seventh virtual server will be created via kickstart in Chapter 9, “Installing Linux with kickstart” on page 149.You will, however, prepare six new user IDs so kickstart will have a user ID to install into.

The following steps are involved:

- “Formatting and label six new DASD” on page 143
- “Defining six more user IDs” on page 144

- “Testing logging on to a new user ID” on page 147
- “Creating six new parameter files” on page 145
- “Granting user IDs access to VSWITCH” on page 146

8.4.1 Formatting and label six new DASD

Decide which DASD will be used for the six new user IDs by referring to 2.7.2, “z/VM DASD worksheet” on page 18.

Logon to MAINT. Query the devices that you want to assign as PERM space. In this example the devices are A741-A746:

```
==> q <a741-a746>
DASD A741 ZA712 , DASD A742 ZA742 , DASD A743 ZA743 , DASD A744 ZA744
DASD A745 ZA745 , DASD A746 ZA746
```

Attach the DASD devices to MAINT. When attaching volumes to your own user ID, the * parameter can be used (so in the following example * is expanded to MAINT):

```
==> att <a741-a746> *
A741-A746 ATTACHED TO MAINT
```

Note: if the DASD are already attached to SYSTEM, you have to first detach them. For example **det a741-a746 system**

Now format the DASD for PERM or minidisk space with the **CPFORMAT** command that is associated with this book.

```
==> cpformat <a741-a746> as perm

Label the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A741 MAINT    A741 3390  ZA741  A741          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A742 MAINT    A742 3390  ZAA742 A742          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A743 MAINT    A743 3390  ZAA743 A743          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A744 MAINT    A744 3390  ZAA744 A744          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A745 MAINT    A745 3390  ZAA745 A745          0       3339
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A746 MAINT    A746 3390  ZAA746 A746          0       3339

ARE YOU SURE you want to format the DASD as PERM space (y/n)?
y
...
A741-A746 DETACHED
A741-A746 ATTACHED TO MAINT

DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A741 MAINT    A741 3390  MMA741 A741          0       3339
MAINT    A742 MAINT    A742 3390  MMA742 A742          0       3339
MAINT    A743 MAINT    A743 3390  MMA743 A743          0       3339
MAINT    A744 MAINT    A744 3390  MMA744 A744          0       3339
MAINT    A745 MAINT    A745 3390  MMA745 A745          0       3339
MAINT    A746 MAINT    A746 3390  MMA746 A746          0       3339
```

You could now shutdown and re-IPL for these new DASD to be attached to the system via the `User_Volume_Include MM*` statement in the `SYSTEM CONFIG` file. However, there is an easier way. Simply **DETACH** the DASD from `MAINT` and **ATTACH** them to `SYSTEM`:

```
==> det <a741-a746>
A741-A746 DETACHED BY MAINT
==> att <a741-a746> system
DASD A741 ATTACHED TO SYSTEM MMA741
DASD A742 ATTACHED TO SYSTEM MMA742
DASD A743 ATTACHED TO SYSTEM MMA743
DASD A744 ATTACHED TO SYSTEM MMA744
DASD A745 ATTACHED TO SYSTEM MMA745
DASD A746 ATTACHED TO SYSTEM MMA746
```

The six DASD volumes will now be available to be used for minidisks in the `USER DIRECT` file. They will also be available after the next IPL because their new labels match the pattern **`User_Volume_Include MM*`** in the `SYSTEM CONFIG` file.

8.4.2 Defining six more user IDs

Define six more user IDs for Linux virtual servers in the `USER DIRECT` file named `LINUX02 - LINUX07`. You will need to use the DASD volumes you just formatted: one for each virtual server. You can repeat the definition of `LINUX01` six times with the block copy **`""6`** prefix command. For example:

```
==> x user direct
====> /user linux01
...
""6   *
01846 USER LINUX01 LNX4VM 256M 1G G
01847  INCLUDE LNXDFLT
01848  OPTION APPLMON
""6    MDISK 100 3390 0001 3338 MMA740 MR LNX4VM LNX4VM LNX4VM
```

This will create six more copies of the `LINUX01`. Modify them to have a user ID of `LINUX02` - `LINUX07`, and give each new ID the proper 3390-3 identified by label (`MMA712-MMA746` in this example):

```
USER LINUX02 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3338 MMA741 MR LNX4VM LNX4VM LNX4VM
*
USER LINUX03 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3338 MMA742 MR LNX4VM LNX4VM LNX4VM
*
...
USER LINUX07 LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3338 MMA746 MR LNX4VM LNX4VM LNX4VM
```

Go to the top of the file and find the definition for the user `$ALLOC$`. Add dummy definitions for cylinder 0 of each of the new volumes and save the changes:

```
====> top
====> /alloc
USER $ALLOC$  NOLOG
 MDISK A01 3390 000 001 520RES R
```

```
...
 MDISK A0B 3390 000 001 MMA740 R
 MDISK A0C 3390 000 001 MMA741 R
 MDISK A0D 3390 000 001 MMA742 R
 MDISK A0E 3390 000 001 MMA743 R
 MDISK A0F 3390 000 001 MMA744 R
 MDISK A10 3390 000 001 MMA745 R
 MDISK A11 3390 000 001 MMA746 R
...
====> file
```

Check for overlaps and the single gap. **`Quit`** out of the `USER DISKMAP` file:

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

Bring the changes online with the **`DIRECTXA USER`** command:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
```

You have now created six new user IDs that can be cloned to.

8.4.3 Creating six new parameter files

New parameter and configuration files must be created for each of the user IDs with the proper networking information. Link and access the `LNXMAINT 192 disk` read/write and create six new parameter files for `LINUX02` through `LINUX06` (`LINUX07` will be configured in Chapter 9, “Installing Linux with kickstart” on page 149)

```
==> link lnxmaint 192 1192 mr
==> acc 1192 f
==> copy linux01 parm-rh5 f linux02 = =
==> copy linux01 parm-rh5 f linux03 = =
==> copy linux01 parm-rh5 f linux04 = =
==> copy linux01 parm-rh5 f linux05 = =
==> copy linux01 parm-rh5 f linux06 = =
```

Now create six new configuration files for `LINUX02` through `LINUX07`

```
==> copy linux01 conf-rh5 f linux02 = =
==> copy linux01 conf-rh5 f linux03 = =
==> copy linux01 conf-rh5 f linux04 = =
==> copy linux01 conf-rh5 f linux05 = =
==> copy linux01 conf-rh5 f linux06 = =
```

Edit each of the six parameter files so they point to the appropriate configuration file

```
==> x linux02 parm-rh5 f
ramdisk_size=40000 root=/dev/ram0 ro ip=off
CMSDASD=191 CMSCONFFILE=LINUX02.CONF-RH5
vnc vncpassword=lnx4vm
==> x linux03 parm-rh5 f
...
```

Edit each of the six configuration files, replacing the appropriate network values. It is likely that only the TCP/IP address and host name will need to be modified:

```
==> x linux02 conf-rh5 f
DASD=100-102
HOSTNAME=lat122.pbm.ihost.com
NETTYPE=qeth
IPADDR=129.40.178.122
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETWORK=129.40.178.0
NETMASK=255.255.255.0
SEARCHDNS=pbm.ihost.com
BROADCAST=129.40.178.255
GATEWAY=129.40.178.254
DNS=129.40.106.1
MTU=1500
PORTNAME=DONTCARE
==> x linux03 conf-rh5 f
...
```

When you are finished, release and detach the `LNXMAINT 192` disk.

```
==> rel f (det
DASD 1192 DETACHED
```

8.4.4 Granting user IDs access to VSWITCH

Modify the `PROFILE EXEC` on `AUTOLOG1 191` to grant access to the VSWITCH for the six new user IDs and add **`XAUTOLOG`** commands so they will booted when the z/VM system IPLs.

Link and access the `AUTOLOG1 191 disk` so the file can be modified from `MAINT`:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
```

Edit the **`PROFILE EXEC`**:

```
==> x profile exec f
...
/* Grant access to VSWITCH for each Linux user */
'CP SET VSWITCH VSW1 GRANT RHEL5'
'CP SET VSWITCH VSW1 GRANT LINUX01'
'CP SET VSWITCH VSW1 GRANT LINUX02'
'CP SET VSWITCH VSW1 GRANT LINUX03'
'CP SET VSWITCH VSW1 GRANT LINUX04'
'CP SET VSWITCH VSW1 GRANT LINUX05'
'CP SET VSWITCH VSW1 GRANT LINUX06'
'CP SET VSWITCH VSW1 GRANT LINUX07'

/* XAUTOLOG each Linux user that should be started */
'CP XAUTOLOG RHEL5'
'CP XAUTOLOG LINUX01'
'CP XAUTOLOG LINUX02'
'CP XAUTOLOG LINUX03'
'CP XAUTOLOG LINUX04'
'CP XAUTOLOG LINUX05'
'CP XAUTOLOG LINUX06'
'CP XAUTOLOG LINUX07'
====> file
```

It is easiest to grant access to the new user IDs for the current z/VM session with the **SET VSWITCH** command:

```
==> set vswitch vsw1 grant linux02
Command complete
==> set vswitch vsw1 grant linux03
Command complete
==> set vswitch vsw1 grant linux04
Command complete
==> set vswitch vsw1 grant linux05
Command complete
==> set vswitch vsw1 grant linux06
Command complete
==> set vswitch vsw1 grant linux07
Command complete
```

Verify that the user IDs have access with the **QUERY VSWITCH ACCESSLIST** command:

```
==> query vswitch vsw1 acc
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 4 Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                Accounting: OFF
  VLAN Unaware
  State: Ready
  IPTimeout: 5        QueueStorage: 8
Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1 VDEV:  3004
  Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2 VDEV:  3008 BACKUP
    Authorized userids:
       RHEL5    LINUX01   LINUX02   LINUX03   LINUX04   LINUX05
       LINUX06  LINUX07
       SYSTEM
```

8.4.5 Testing logging on to a new user ID

You should now be able to logon to a new user ID and verify the integrity of the definitions. **Logon to LINUX02** and you should first notice that a NIC is created as well as two VDISKs:

```
LOGON  LINUX02
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 10:17:55 EST WEDNESDAY 12/13/06
z/VM V5.2.0    2006-07-18 08:48


DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 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, :

```
==> q osa
00: OSA  0600 ON NIC  0600  UNIT 000 SUBCHANNEL = 0002
00:      0600 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
...
```

Verify that you have three read/write devices at addresses 100-102 via the **QUERY DASD** command:

```
==> q da
00: DASD 0100 3390 MMA741 R/W       3038 CYL ON DASD  A719 SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W     524288 BLK ON DASD  VDSK SUBCHANNEL = 000E
00: DASD 0102 9336 (VDSK) R/W    1048576 BLK ON DASD  VDSK SUBCHANNEL = 0001
00: DASD 0190 3390 MVA700 R/O        107 CYL ON DASD  A700 SUBCHANNEL = 0009
...
```

Logoff of LINUX02.

Congratulations, you have cloned one Linux virtual server and defined six more user IDs that should now be ready for cloning or kickstarting to. You will clone to these user IDs in the chapters that follow.

8.5 Reviewing system status

You can step back now and view your system from a DASD point of view as shown in Figure 8-2. If you have followed all sections in this book you should have used 24 3390-3 volumes: 10 for your z/VM system, 7 for the Linux controller and master image and one for each of the seven virtual servers.

You can also view the system from an administrator's and end user point of view as shown by the horizontal lines and the italicized text on the right side of the figure. The z/VM and Linux system administration roles may be performed by the same person, but these roles can also be done by different administrators. The Linux end users may not care that their servers are virtual machines and may be oblivious to the fact that they might have been cloned in a matter of minutes.

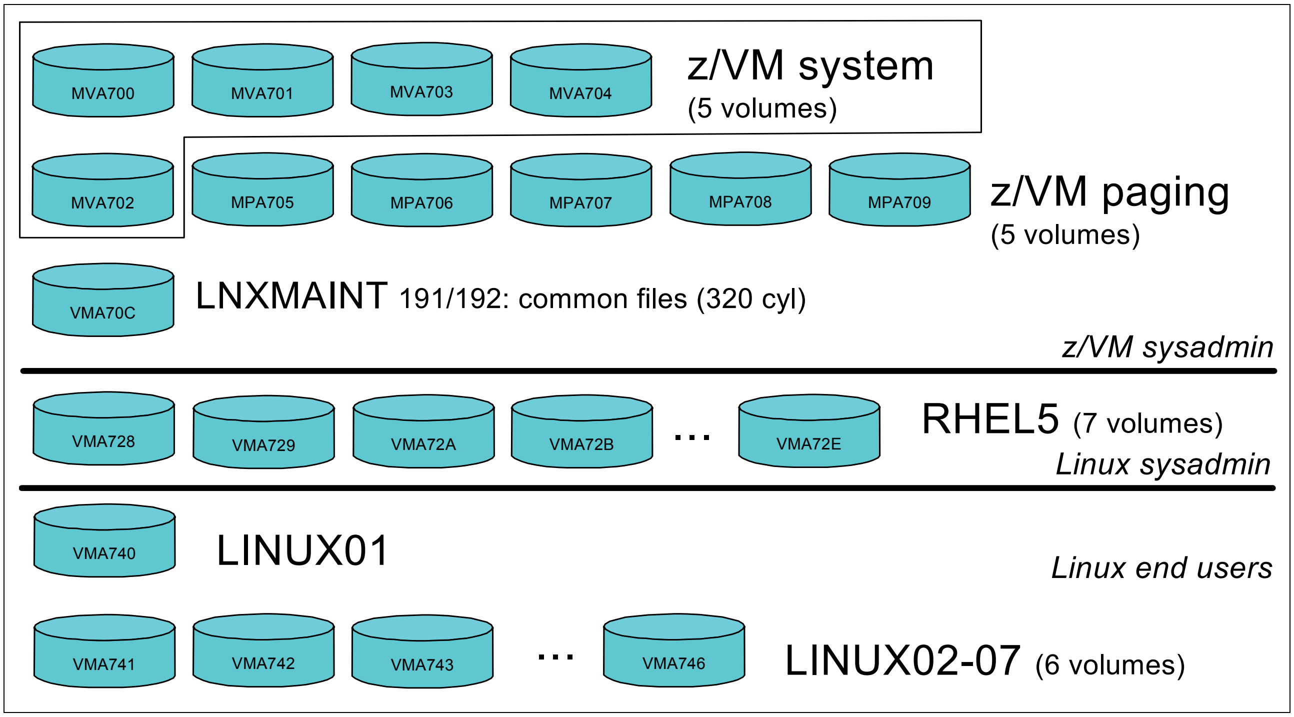


Figure 8-2 Linux virtual server system - DASD view and role view

Installing Linux with kickstart

Kickstart is an automated way of installing RHEL 5. Using kickstart, you can create a single file that answers all of the questions usually asked during an interactive installation.

In the last chapter, you cloned six new virtual servers, and in this chapter you will kickstart a seventh. In comparison, cloning a server is faster, assuming the FLASHCOPY command is available. However, kickstarting a server is more flexible, as it allows for different package configurations as well as pre-install and post-install scripting.

The controller is now configured as an installation server, with the installation tree shared via NFS. You will now configure it as a kickstart server to perform automated installations over the network. You will then install the final Linux server with kickstart.

The following steps are involved in installing Linux with kickstart:

- Configure the controller for kickstart
- Configure the LINUX07 user for kickstart
- Kickstart the LINUX07 user

9.1 Configure the controller for kickstart

In this section, you configure the controller to host the kickstart file, which you use to perform the automated installation of the Linux virtual server called LINUX07. This section assumes you have already set up the RHEL 5 install tree as described in 7.7, “Configuring the controller” on page 121.

The installer generates a kickstart file at the end of every installation. It is based on the answers provided during the interactive install. This kickstart file is named anaconda-ks.cfg and is located in the /root/ directory. You use this kickstart file from RHEL5 as a template for LINUX07. Perform the following steps:

1. From an Secure Shell (SSH) session to the controller, create a directory to hold the kickstart file. Then, copy the default kickstart file to this directory, rename it, and modify the permissions so others may read it:

```
# mkdir /nfs/ks
# cp /root/anaconda-ks.cfg /nfs/ks/linux07-ks.cfg
# chmod +r /nfs/ks/linux07-ks.cfg
```

2. Next, use **vi** to edit the kickstart configuration file. Remove the lines with the ~~strike-through~~, and edit the lines in bold in to customize this kickstart for LINUX01:

```
# vi /nfs/ks/linux07-ks.cfg
install
nfs --server=129.40.178.128 --dir=/nfs/rhel5                                        1
lang en_US.UTF-8
network --device eth0 --mtu=1500 --bootproto static --ip 129.40.178.127             2
--netmask 255.255.255.0 --gateway 129.40.178.254 --nameserver 129.40.106.1
--hostname linux07.pbm.ihost.com                                                    3
rootpw --iscrypted $1$d6Mi5lj9$jzwhlKOUK37qXQzTLVRSm.
firewall --disabled                                                                 4
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone America/New_York
bootloader --location=mbr ~~--driveorder=dasda,dasdd,dasdg,dasdh,dasdi,dasdj,dasdk~~   5
reboot                                                                              6
key --skip                                                                          7
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
zerombr yes                                                      8
clearpart --all --initlabel                                      9
part / --fstype ext3 --size=1 --grow --ondisk=dasda             10
~~#part pv.6 --noformat --onpart dasdg1~~                           11
~~#part pv.5 --noformat --onpart dasdf1~~
~~#part pv.3 --noformat --onpart dasdd1~~
~~#part pv.4 --noformat --onpart dasde1~~
~~#volgroup VolGroup00 --pesize=32768 pv.3 pv.4 pv.5 pv.6~~
~~#logvol /nfs --fstype ext3 --name=LogVol00 --vgname=VolGroup00 --size=8992~~

%packages                                                       12
@base
@base-x
@core
comps-extras
cracklib-dicts
gnome-mime-data
kernel-headers
nash
rmt
tzdata

%post                                                           13
echo ARP=no >> /etc/sysconfig/network-scripts/ifcfg-eth0        14
echo /dev/dasdb1 swap swap defaults 0 0 >> /etc/fstab           15
echo /dev/dasdc1 swap swap defaults 0 0 >> /etc/fstab
```

Note the following points, which explain the command lines that are numbered above:

1 IP address of installation server and path to install tree.

2 IP address of new linux server

3 Host name of new linux server

4 Disable firewall. Only advisable if server is not on an external network.

5 Remove references to additional drives only available to the controller.

6 Add this line, so that the server automatically shuts down after kickstart

7 Skip entering an installation number. If you have one, enter it here instead of using **`--skip`**. For more information, see “Stage 2 of the RHEL 5 installation” on page 99.

8 Add this line to perform a non-interactive direct access storage device (DASD) format.

9 Removes all existing partitions. Uncomment this line and add --initlabel

10 Defines the root partition. The `--size 1` and `--grow` options specify all of the 100 disk.

11 Previously defined Logical Volume Manager (LVM) used for install tree. Delete these lines.

12 Specifies packages to install. The `@core` entry is required for a minimal install.

13 Add section for post-install script.

14 This step is performed after installation. `ARP=no` is required for interfaces on a layer 3 z/VM Virtual Switch (VSWITCH).

15 Add entries for virtual disk (VDISK) swap spaces in `/etc/fstab`.

> **Note:** The zerombr and clearpart options mentioned here are chosen because the DASD are not yet formatted for Linux use. For future kickstarts to the same DASD volumes, they will already be formatted. Thus, you can save a substantial amount of time by only recreating the file systems and not reformatting the DASD. In this case you would use the following kickstart options:
>
> ```
> zerombr no
> clearpart --all
> ```

3. Finally, add the path to the kickstart folder to `/etc/exports`

```
# vi /etc/exports
/nfs/rhel5/             *(ro,sync)
/nfs/virt-cookbook-RH5  *(ro,sync)
/nfs/ks                 *(ro,sync)
```

4. Restart the NFS service on the controller. The **`showmount -e`** command should show the exported file systems:

```
# service nfs restart
Shutting down NFS mountd: [  OK  ]
Shutting down NFS daemon: [  OK  ]
Shutting down NFS quotas: [  OK  ]
Shutting down NFS services:  [  OK  ]
Starting NFS services:  [  OK  ]
Starting NFS quotas: [  OK  ]
Starting NFS daemon: [  OK  ]
Starting NFS mountd: [  OK  ]
# showmount -e
Export list for lat128.pbm.ihost.com:
/nfs/ks               *
/nfs/rhel5            *
/nfs/virt-cookbook-RH5 *
```

9.2 Configure the LINUX07 user for kickstart

Earlier you should have created the user ID LINUX07, it is time to configure it for kickstart. `LINUX07` must have its own parameter and configuration files, which are again based on the `RHEL5` user ID.

`LOGOFF` of `MAINT` and logon to `LNXMAINT`. Copy the parameter and configuration files from `RHEL5` to `LINUX07` as follows:

```
==> copy rhel5 parm-rh5 d linux07 = =
==> copy rhel5 conf-rh5 d linux07 = =
```

Edit the `LINUX07 PARM-RH5` file. Because this is a non-interactive installation, the vnc options are no longer required. The **`ks=`** line directs the installer to get the kickstart file from the installation server. **`RUNKS=1`** is required for kickstarts, and the **`cmdline`** option prevents the installer's text-based user interface from opening on the 3270 console:

```
==> x linux07 parm-rh5 d
ramdisk_size=40000 root=/dev/ram0 ro ip=off
CMSDASD=191 CMSCONFFILE=linux07.conf-rh5
ks=nfs:129.40.178.128:/nfs/ks/linux07-ks.cfg
RUNKS=1 cmdline
====> file
```

Next, edit the `LINUX07 CONF` file, and change the DASD range and networking information:

```
==> x linux07 conf-rh5 d
DASD=100-102
HOSTNAME=linux07.pbm.ihost.com
NETTYPE=qeth
IPADDR=128.40.178.127
...
====> file
```

9.3 Kickstart the LINUX07 user

Perform the following steps to kickstart the `LINUX07` user:

1. Logoff of `LNXMAINT` and logon to `LINUX07`. When asked to IPL from disk 100, answer `n`:

```
   LOGON  LINUX07
   ...
   DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
   Do you want to IPL Linux from DASD 100? y/n
   n
00:      0602 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
```

2. As in "Installing the master image" on page 95, we will need to temporarily add more memory for the install process. Temporarily modify the storage up to 512MB with the **`DEFINE STORAGE`** command. Then **`IPL CMS`** and again answer **n** to the question of IPLing Linux:

```
==> def stor 512m
00: STORAGE = 512M
00: Storage cleared - system reset.
==> ipl cms
...
Do you want to IPL Linux from DASD 200? y/n
n
```

Verify that you have a 512MB virtual machine:

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

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

3. Run **`rhel5 exec`** to initiate the kickstart. You see some initial kernel messages, followed by the file system format and Red Hat Package Manager (RPM) package installation.

   Note: Towards the end of the kickstart, it is normal to see some unrecognized characters on the screen. This is because the 3270 console cannot display the progress meter during the post installation phase. To automatically clear the 3270 console and avoid multiple screens of unreadable messages, issue the **`#cp term more 0 0`** command before running **`RHEL5 EXEC`**.

   ```
   ==> rhel5
   ...
   Kernel command line: ramdisk_size=40000 root=/dev/ram0 ro ip=off
                        CMSDASD=191 CMSCONFFILE= LINUX07.conf
                        ks=nfs:129.40.178.128:/nfs/ks/linux07-ks.cfg
                        RUNKS=1 cmdline
   ...
   ```

4. The first time kickstart is run, the installer must format the DASD for Linux use. It is normal to see error messages of the following format if the DASD you are using has never been formatted. In subsequent kickstart installs, you should not see these errors:

   ```
   end_request: I/O error, dev dasda, sector 0
   Buffer I/O error on device dasda, logical block 0
   Please wait while formatting drive dasda...
   ```

   Note: The VDISK swap device is configured in the %post section of the kickstart file.

5. At the end of the kickstart, IPL the 100 disk to make any changes to your RHEL 5 master image:

   ```
    /mnt/sysimage/dev done
    /mnt/sysimage done
   you may safely reboot your system
   ==> #cp ipl 100
   00: zIPL v1.5.3 interactive boot menu
   00:  0. default (linux)
   00:  1. linux
   ...
   ```

Congratulations! You have now installed Linux onto the virtual server using kickstart. This process can be repeated in the future for other Linux guests. For the purpose of this book, we present a minimal installation with kickstart. However, you can completely customize the kickstart file to install different packages based on your requirements. For more information regarding kickstart options, see the documentation located at:

http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/

10

Servicing Linux with Red Hat Network

This chapter describes Red Hat Network (RHN) and its ability to manage the virtual servers. Using **yum**, the virtual servers can be updated when Red Hat errata are released. You can also use **yum** to install new packages with automatic dependency resolution. RHN is accessed by the following link:

http://rhn.redhat.com/

The following sections describe how to configure a Linux guest for **yum**, and manage the guest through RHN:

- “Registering your system with RHN” on page 155.
- “Installing and updating packages using yum” on page 156.
- “Managing your Linux guest through RHN” on page 157.
- “Updating a system without Internet access” on page 158.

10.1 Registering your system with RHN

This section assumes you have already obtained a valid entitlement for RHEL 5 on System z, or have completed the steps to obtain an evaluation copy. To receive a free 90-day evaluation, visit:

http://www.redhat.com/rhel/details/eval

Select the link named **Red Hat Enterprise Linux AS for IBM eServer zSeries and IBM S/390** and create an account. Before using **yum** for the first time, you must import the Red Hat GPG key and register your Linux guest with RHN. Use the commands below, substituting your RHN user name, password, and host name of the Linux guest.

```
# rpm --import /usr/share/rhn/RPM-GPG-KEY
# rhnreg_ks --username=myuser --password=mypw --profilename=linux01.pbm.ihost.com
```

10.2 Installing and updating packages using yum

Now that your system is registered with RHN, you can use **yum** to keep the system updated. You can download and install the latest version of a package by running **yum** with the RPM package name. You can also specify multiple packages on the command line separated by spaces. The **yum install** command installs the package if it is not present, and the **yum upgrade** command updates to the latest version if it is already installed. If a package has any dependencies, **yum** automatically downloads and installs them for you.

Update the cpp package to get the latest security fixes:

```
# rpm -q cpp
cpp-4.1.1-30
# yum upgrade cpp
Loading "rhnplugin" plugin
Loading "installonlyn" plugin
Setting up Upgrade Process
Setting up repositories
rhel-s390x-server-5-beta  100% |=========================|  950 B    00:00
...
==============================================================================
 Package                 Arch       Version           Repository        Size
==============================================================================
Updating:
 cpp                     s390x      4.1.1-43.el5      RHEL5             2.6 M

Transaction Summary
==============================================================================
Install      0 Package(s)
Update       1 Package(s)
Remove       0 Package(s)

Total download size: 2.6 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : cpp                           ######################### [1/2]
  Cleanup   : cpp                           ######################### [2/2]

Updated: cpp.s390x 0:4.1.1-43.el5
Complete!
```

Now query the cpp package and you should see that it has been updated.

```
# rpm -q cpp
cpp-4.1.1-43.el5
```

To update every installed package on the system, run:

```
# yum upgrade
```

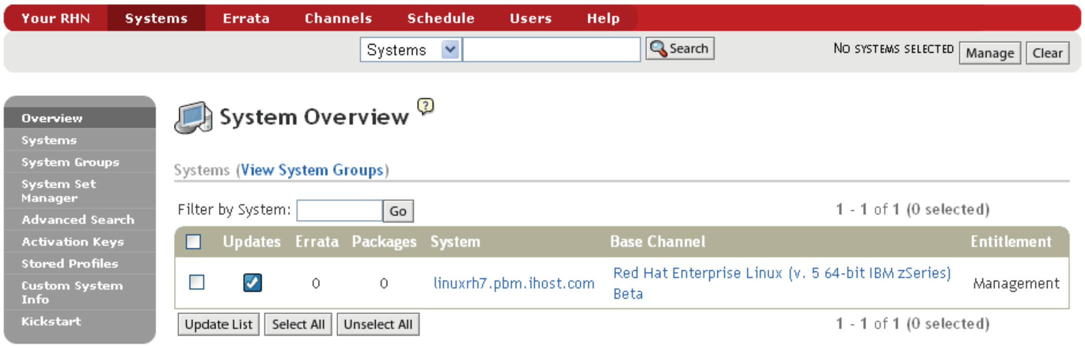
For more information about the **yum** command see the yum(8) man page.

10.3 Managing your Linux guest through RHN

You can also manage the packages on this Linux guest through the Web interface at:

```
http://rhn.redhat.com/
```

When you first log in to RHN, you see the system you registered under the **Systems** tab. If there is a red exclamation point next to your system, there are errata waiting to be applied. The number of relevant errata and the corresponding number of packages are visible to the left of the system name. Click the number beneath Errata or Packages to get a detailed list. If there is a blue check-mark, then the system is fully updated.



System Overview

Systems (View System Groups)

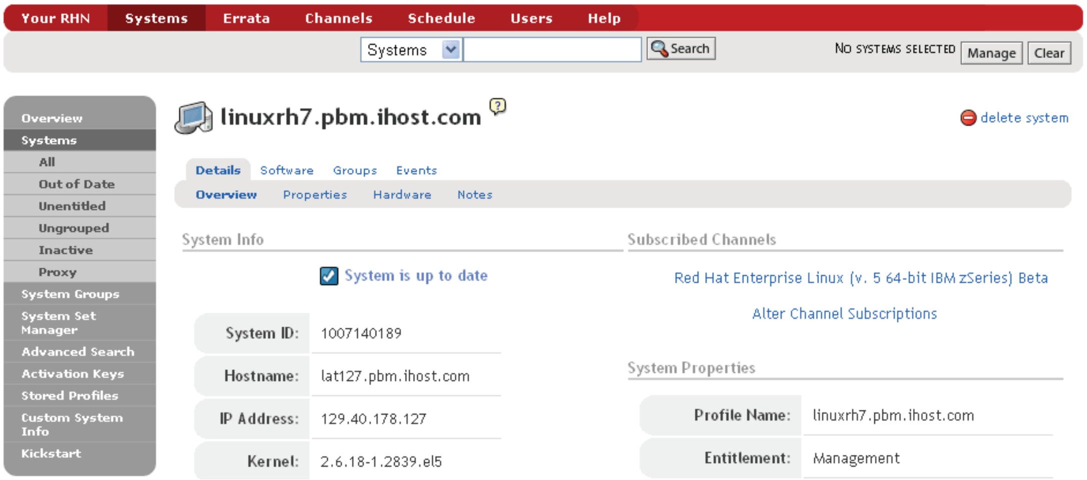
Filter by System: Go — 1 - 1 of 1 (0 selected)

□	Updates	Errata	Packages	System	Base Channel	Entitlement
□	☑	0	0	linuxrh7.pbm.ihost.com	Red Hat Enterprise Linux (v. 5 64-bit IBM zSeries) Beta	Management

Update List | Select All | Unselect All — 1 - 1 of 1 (0 selected)

Figure 10-1 RHN system overview

Next, click the link that is the system name. This brings you to a detailed overview, where you can see the system properties as Figure 10-2 shows. Click the Packages tab to view all packages installed on this system. From this tab, you can also update, remove, or install new packages onto the system.



linuxrh7.pbm.ihost.com

System Info

☑ System is up to date

System ID:	1007140189
Hostname:	lat127.pbm.ihost.com
IP Address:	129.40.178.127
Kernel:	2.6.18-1.2839.el5

Subscribed Channels

Red Hat Enterprise Linux (v. 5 64-bit IBM zSeries) Beta

Alter Channel Subscriptions

System Properties

Profile Name:	linuxrh7.pbm.ihost.com
Entitlement:	Management

Figure 10-2 RHN system details

For more information about managing your systems through RHN, including usage guides and frequently asked questions, see:

```
http://rhn.redhat.com/help
```

10.4 Updating a system without Internet access

The **yum** command communicates with RHN using secure Web traffic. If your Linux guest is not connected to the Internet, or is behind a firewall that prevents access to RHN, you can still use all of the **yum** functionality.

10.4.1 Configuring the server

This process uses the controller configured Chapter 7., "Installing and configuring RHEL 5" as a **yum** repository. The controller exports the RHEL 5 install tree over NFS. Your Linux guest uses the automounter to access this install tree, which **yum** uses instead of RHN.

For more information on exporting the install tree over NFS, refer back to 7.7.8, "Turning on the NFS server" on page 125. The following sections assume the install tree is exported over NFS as `/nfs/rhel5/`.

10.4.2 Configuring the client

On the Linux guest, you will first configure the automounter to mount the installation tree from the controller on demand. The automounter will automatically mount a remote directory when it is accessed, and automatically unmount it after a period of inactivity. Edit the file `/etc/auto.master` and add the following line at the bottom:

```
# vi /etc/auto.master
...
/nfs    /etc/auto.controller
```

Next, create the file `/etc/auto.controller`, and add the following line, substituting the IP address of your controller:

```
# vi /etc/auto.controller
rhel5           -ro,hard,intr   <129.40.178.128>:/nfs/rhel5
```

Now, create the `/nfs` directory. Restart the **autofs** service to pick up the configuration changes, then list the contents of the automounted directory:

```
# mkdir /nfs
# service autofs restart
Stopping automount: [  OK  ]
Starting automount: [  OK  ]
# ls /nfs/rhel5
EULA            README-or.html          RELEASE-NOTES-ja.html
eula.en_US      README-pa.html          RELEASE-NOTES-ko.html
...
```

Now that the installation tree is accessible, you will configure **yum** to use it. Create a file named `rhel5.repo` in the `/etc/yum.repos.d/` directory:

```
# vi /etc/yum.repos.d/rhel5.repo
[RHEL5]
name=Red Hat Enterprise Linux 5
baseurl=file:///nfs/rhel5/Server
```

Finally, you need to import the RPM GPG key so that **yum** knows you are installing official Red Hat packages. The Red Hat GPG key is located in the install tree. Import the key with:

```
# rpm --import /nfs/rhel5/RPM-GPG-KEY
```

Note: Red Hat signs each RPM with a private GPG key, which is compared to your public key each time a package is installed. This method ensures that the RPM is a genuine, unaltered package. When installing an RPM, if you ever see a message similar to:

```
Header V3 DSA signature: NOKEY, key ID 897da07a
```

Either the correct GPG key has not been imported, or the package itself has been altered by a third party.

You are now ready to use **yum** to install or upgrade an RPM package. Refer back to "Installing and updating packages using yum" on page 156 for **yum** syntax.

10.4.3 Updating packages on the server

The RPMs are located in the /Server subdirectory of the install tree. Beneath this directory, there is a subdirectory named /repodata which contains repository files used by **yum**. Whenever a new set of RPMs is added to the /Server directory, you must rebuild the repository data with the **createrepo** command.

On the controller, first install the **createrepo** RPM with:

```
# yum install createrepo
...
Running Transaction
  Installing: createrepo                 ######################### [1/1]

Installed: createrepo.noarch 0:0.4.4-2.fc6
Complete!
```

Next, backup the current repository and create a new one with the **createrepo** command:

```
# cd /nfs/rhel5/Server
# mv repodata repodata.orig
# createrepo /nfs/rhel5/Server
96/2480 - system-config-network-1.3.97-1.el5.noarch.rpm
...
```

It should take about five minutes to build the new repository data, depending on the number of RPMs.

You now have a new repository based on the RPMs in /nfs/rhel5/Server. You should repeat this procedure any time an RPM (or set of RPMs) change. You can also use this to create multiple RPM repositories, based on anything you like. For example, you can have a repository named “test” containing the latest packages from RHN, as well as a “production” repository based on an official RHEL update. You can then create a .repo file on the client pointing to either repository, based on the role of that client.

For example, on the server:

```
# cd /nfs/rhel5
# cp -a Server Server.production
# cp -a Server Server.test
```

Now, any time you update an RPM in /nfs/rhel5/Server.test/, run **createrepo** on that directory. On the client, in the /etc/yum.repos.d/ directory you can have either:

```
/etc/yum.repos.d/rhel5.production.repo:
[RHEL5]
name=RHEL 5 production
baseurl=file:///nfs/rhel5/Server.production/
```

or

```
/etc/yum.repos.d/rhel5.test.repo:
[RHEL5]
name=RHEL 5 test
baseurl=file:///nfs/rhel5/Server.test/
```

Note that this method requires maintaining separate copies of each directory of RPMs, which can fill the /nfs directory on the controller rather quickly. Red Hat provides a separate product named Satellite which maintains a single tree, while managing the RPM version differences. The Satellite provides other features similar to RHN as well. For more information, see:

https://www.redhat.com/rhn/rhndetails/architecture/

11

Cloning open source virtual servers

The secret to creativity is knowing how to hide your sources.

--Albert Einstein

This chapter describes how to clone and customize the following Linux virtual servers:

- “Creating a virtual Web server” on page 161
- “Creating a virtual LDAP server” on page 165
- “Creating a virtual file and print server” on page 173
- “Creating a virtual application development server” on page 177

The sections that follow don’t go into the theory nor detail on the four types of servers. Rather, they are just a reference to get the servers quickly installed and configured. There are many other resources that go into depth on these types of servers.

11.1 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 8, “Configuring RHEL 5 for cloning” on page 133.

11.1.1 Installing Apache RPMs

SSH into the IP address of the new `LINUX01` server. Install the following Apache RPMs via the **`yum -y install`** command. The **`-y`** flag prevents the “Is this OK” question:

```
# yum -y install httpd httpd-manual
...
==============================================================================
 Package                 Arch       Version           Repository        Size
==============================================================================
Installing:
 httpd                   s390x      2.2.3-6.el5       RHEL5             1.1 M
 httpd-manual            s390x      2.2.3-6.el5       RHEL5             831 k
Installing for dependencies:
 apr                     s390x      1.2.7-10          RHEL5             122 k
 apr-util                s390x      1.2.7-3          RHEL5              78 k
 postgresql-libs         s390x      8.1.4-1.1        RHEL5             197 k

Transaction Summary
=============================================================================
Install      5 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 2.3 M
Downloading Packages:
Running Transaction Test
warning: httpd-manual-2.2.3-6.el5: Header V3 DSA signature: NOKEY, key ID 897da07a
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: apr                         ######################### [1/5]
  Installing: postgresql-libs             ######################### [2/5]
  Installing: apr-util                    ######################### [3/5]
  Installing: httpd                       ######################### [4/5]
  Installing: httpd-manual                ######################### [5/5]

Installed: httpd.s390x 0:2.2.3-6.el5 httpd-manual.s390x 0:2.2.3-6.el5
Dependency Installed: apr.s390x 0:1.2.7-10 apr-util.s390x 0:1.2.7-3
postgresql-libs.s390x 0:8.1.4-1.1
Complete!
```

Verify that the RPMs were installed

```
# rpm -qa | grep httpd
httpd-2.2.3-6.el5
httpd-manual-2.2.3-6.el5
```

Before starting the Apache Web server, use the **chkconfig** command to set the service to start at boot time:

```
# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
# chkconfig httpd on
# chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

11.1.2 Testing Apache

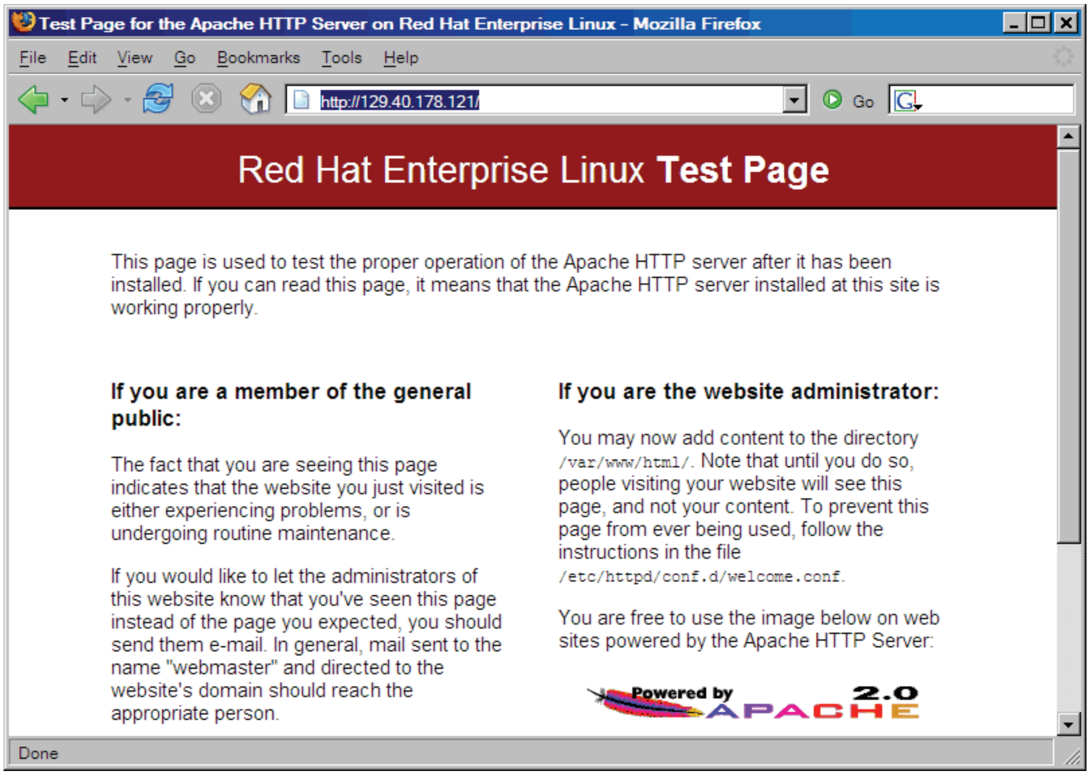
Start the Apache Web server to verify it is installed successfully.

```
# service httpd start
Starting httpd: [  OK  ]
```

To verify that Apache is installed correctly, after it’s been started, **bring up a Web browser** and point it to the server. For example, the virtual server running on LINUX01 can be reached with the following URL:

```
http://<129.40.178.121>/
```

You should see the following test page to verify the Web server is working:


Test Page for the Apache HTTP Server on Red Hat Enterprise Linux - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.121/
Go
Done

Red Hat Enterprise Linux **Test Page**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Figure 11-1 Apache test page

If you get an error in starting Apache, look in the log file `/var/log/httpd/error-log` for clues. If Apache started successfully but you can’t reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

11.1.3 Turning on a firewall

RHEL 5 comes with an IP tables firewall which is disabled after a default installation. Your enterprise may have firewalls in front of Web servers, but you may also want a firewall running on this virtual server. This section describes how to quickly enable an IP tables firewall and configure it to allow Web traffic through.

Verify that the firewall is off via the **`chkconfig --list`** command. The service name is **`iptables`**:

```
# chkconfig --list iptables
iptables        0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

Turn on the firewall at boot time with the **`chkconfig`** command, and for this session with the **`service`** command:

```
# chkconfig iptables on
# service iptables start
Applying iptables firewall rules: [  OK  ]
Loading additional iptables modules: ip_conntrack_netbios_ns [  OK  ]
```

Go back to your browser and **click refresh**. You should get an error that the server is not responding. This is because packets for ports for http: and https: (80 and 443) are dropped by default. To allow Web traffic through, you can modify the file `/etc/sysconfig/iptables`. First make a backup copy, then add two rules (emboldened) to allow these ports then save your changes:

```
# cd /etc/sysconfig
# cp iptables iptables.orig
# vi iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Restart the firewall to pick up the new rules:

```
# service iptables restart
Flushing firewall rules: [  OK  ]
Setting chains to policy ACCEPT: filter [  OK  ]
Unloading iptables modules: [  OK  ]
Applying iptables firewall rules: [  OK  ]
Loading additional iptables modules: ip_conntrack_netbios_ns [  OK  ]
```

Go back to your browser and **click refresh** again. You should not get an error this time. You should now have a firewall that allows Web traffic.

11.1.4 Configuring SSL for Apache

Use the Secure Sockets Layer (SSL) to encrypt data between the client (browser) and the server. This is done by specifying an **https** prefix in the URL which uses port 443 rather than using the conventional **http** prefix which uses port 80.

To use SSL, the mod_ssl package is requied. You can show that SSL communications do not work by changing `http` to **https** in your browser:

```
https://<129.40.178.121>/
```

Click reload in your browser and you should get a communications error.

Installing the SSL Apache module

RHEL 5 makes it very easy to install and enable SSL support for Apache. Install the mod_ssl RPM via the **`yum -y install`** command:

```
# yum -y install mod_ssl
...
Install      2 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 212 k
```

```
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: distcache                    ######################## [1/2]
  Installing: mod_ssl                      ######################## [2/2]

Installed: mod_ssl.s390x 1:2.2.3-6.el5
Dependency Installed: distcache.s390x 0:1.4.5-14.1
Complete!
```

Verify that the RPM was added:

```
# rpm -qa | grep ssl
openssl-0.9.8b-8.3.el5
openssl-0.9.8b-8.3.el5
mod_ssl-2.2.3-6.el5
```

Restart the Web server:

```
# service httpd restart
Stopping httpd: [  OK  ]
Starting httpd: [  OK  ]
```

Go back to your browser and click restart again. This time you should get a warning about a self-signed certificate. This is acceptable - for a production Web site you will want a certificate signed by a certificate authority.

11.1.5 Populating your Web site

You can begin to put your Web pages in the directory /var/www/html/ which is the default Web root.

11.1.6 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
```

11.2 Creating a virtual LDAP server

The Lightweight Directory Access Protocol (LDAP) is commonly implemented via 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 install Linux manually and set up login authentication to 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.

The steps in this section are as follow:

- “Cloning a Linux virtual server” on page 166
- “Installing the openldap server” on page 166

- “Configuring the OpenLDAP server” on page 167
- “Migrating existing users to LDAP” on page 168
- “Configuring an LDAP client” on page 171
- “Adding new user accounts” on page 172

11.2.1 Cloning a Linux virtual server

Start an SSH session as root to the controller and clone the master image to the LINUX02 user ID:

```
# clone.sh linux02
Invoking CP command: QUERY LINUX02
HCPCQU045E LINUX02 not logged on
Error: non-zero CP response for command 'QUERY LINUX02': #45
Setting device 0.0.0191 offline
Done
Setting device 0.0.0191 online
Done
LINUX02  CONF-RH5 D1 F         80         13          1 12/22/2006 13:19:54

WARNING!!: this will copy disk 100 to LINUX02 100
New host name will be:  lat182.pbm.ihost.com
New TCP/IP address will be: 129.40.178.182
Other network data is retrieved from LINUX02 CONF-RH5 on 191 disk
Are you sure you want to overwrite these disks (y/n): y
Please enter a brief description of this server
This is an LDAP server running RHEL5
...
Successfully cloned /rhel5master to LINUX02
You should be able to ping 129.40.178.122 within one minute
```

You should now have a vanilla RHEL5 system running on `LINUX02`.

11.2.2 Installing the openldap server

Start an SSH session to the IP address of the new virtual server running on LINUX02. Use the **yum** command to install the OpenLDAP client and server RPMs:

```
# yum -y install openldap-clients openldap-servers
...
==============================================================================
 Package                 Arch       Version          Repository        Size
==============================================================================
Installing:
 openldap-clients        s390x      2.3.27-4         RHEL5             189 k
 openldap-servers        s390x      2.3.27-4         RHEL5             2.3 M
Installing for dependencies:
 libtool-ltdl            s390x      1.5.22-6.1       RHEL5              37 k

Transaction Summary
==============================================================================
Install      3 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 2.5 M
Downloading Packages:
Running Transaction Test
warning: openldap-clients-2.3.27-4: Header V3 DSA signature: NOKEY, key ID 897da07a
Finished Transaction Test
```

```
Transaction Test Succeeded
Running Transaction
  Installing: libtool-ltdl                ######################### [1/3]
  Installing: openldap-clients            ######################### [2/3]
  Installing: openldap-servers            ######################### [3/3]

Installed: openldap-clients.s390x 0:2.3.27-4 openldap-servers.s390x 0:2.3.27-4
Dependency Installed: libtool-ltdl.s390x 0:1.5.22-6.1
Complete!
```

11.2.3 Configuring the OpenLDAP server

Any detailed description of LDAP is outside the scope of this book. Rather, short configuration recommendations are given in this section.

There are two important configuration values that must be chosen.

- The suffix or base distinguished name of the LDAP Domain Information Tree (DIT) - the most common suffix is to use your company's DNS name.
- The LDAP administrator or root name and password.

Choose an administrative password and run the **slappasswd** command which displays an encrypted version of it. The output of this command will be used shortly in a configuration file so you may want to make a copy of it.

```
# slappasswd
New password: <lnx4vm>
Re-enter new password: <lnx4vm>
{SSHA}4FiGwLm+cy+I96TyiWMn4evNXSXa5aJ2
```

The OpenLDAP server configuration file is /etc/openldap/slapd.conf file. Move down about a page to the lines beginning with **suffix** and **rootdn** and set them appropriately. In this example the following values are used:

LDAP suffix	**dc=pbm,dc=ihost,dc=com**
Root user (rootdn)	**cn=ldaproot,dc=pbm,dc=ihost,dc=com**
Root password	**lnx4vm** (encrypted output from **slappasswd**)

Make a backup copy of the main OpenLDAP configuration file, /etc/openldap/slapd.conf, and modify the original. First set the suffix and root user distinguished name:

```
# cd /etc/openldap
# cp slapd.conf slapd.conf.orig
# vi slapd.conf
...
database        bdb
# suffix                "dc=my-domain,dc=com"
# rootdn                "cn=Manager,dc=my-domain,dc=com"
suffix "dc=pbm,dc=ihost,dc=com"
rootdn "cn=ldaproot,dc=pbm,dc=ihost,dc=com"
...
```

Set the **rootpw** line to a value of the encrypted password that was the output of the **slappasswd** command:

```
...
# rootpw                secret
# rootpw                {crypt}ijFYNcSNctBYg
rootpw {SSHA}4FiGwLm+cy+I96TyiWMn4evNXSXa5aJ2
...
```

Save the slapd.conf file. Your LDAP server should now be minimally configured.

Start the LDAP service

Start LDAP at boot time via the **chkconfig** command and for this session via the **service** command:

```
# chkconfig ldap on
# service ldap start
Checking configuration files for slapd:  config file testing succeeded         [  OK  ]
Starting slapd: [  OK  ]
```

Query the LDAP database via the **ldapsearch** command. The **-x** flag specifies that simple authentication is used:

```
# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 32 No such object
```

The result shows that the LDAP directory can be searched, but that it is empty. This is expected as no data has been added to it.

11.2.4 Migrating existing users to LDAP

A common method of populating the LDAP database is via LDAP Directory Interchange Format (LDIF) files. Red Hat includes a set of migration scripts with the OpenLDAP server RPM, written by PADL Software Pty Ltd (http://padl.com). These scripts allow you to convert users and groups from the /etc/ directory to LDIF format. The scripts are in the /usr/share/openldap/migration/ directory. Before converting the /etc/group and /etc/passwd files, add a user that will be used to test LDAP authenticaiton. In this example, **ldapuser1** is added via the **useradd** command and the password is set via the **passwd** command:

```
# useradd ldapuser1
# passwd ldapuser1
Changing password for user ldapuser1.
New UNIX password: <lnx4vm>
Retype new UNIX password: <lnx4vm>
passwd: all authentication tokens updated successfully.
```

The PADL tools have a shared configuration file named migrate_common.ph. Make a backup copy of it and modify it. There are two lines to be changed which set the domain name and suffix (or root) of LDAP tree. In this example the suffix is **pbm.ihost.com**:

```
# cd /usr/share/openldap/migration
# cp migrate_common.ph migrate_common.ph.orig
# vi migrate_common.ph
...
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "<pbm.ihost.com>";

# Default base
```

```
$DEFAULT_BASE = "<dc=pbm,dc=ihost,dc=com>";
...
```

The first tool conversion Perl script is **migrate_base.pl**. Redirect the output to a file named accounts.ldif. This will be the file that you use to populate the LDAP server. After you run it, take a look at the first 9 lines of the LDIF file via the **head** command to give you an idea of the format:

```
# ./migrate_base.pl > accounts.ldif
# head -9 accounts.ldif
dn: dc=ihost,dc=com
dc: ihost
objectClass: top
objectClass: domain

dn: dc=pbm,dc=ihost,dc=com
dc: pbm
objectClass: top
objectClass: domain
```

Important: There appears to be a bug in migrate_base.pl when the suffix has more than two components. This will later result in an error while trying to import the LDIF file:

```
slapadd: line 5: database (dc=pbm,dc=ihost,dc=com) not configured to hold
"dc=ihost,dc=com"
```

If your suffix (Base DN) has more than two components, you can work around the bug by deleting the entry or entries in the LDIF file with fewer components than your suffix. In this example, pbm.ihost.com has three components, so the first entry (first four lines and one blank line) in the LDIF file must be deleted. Then the **slapadd** command succeeded.

The next two scripts migrate the /etc/passwd and /etc/group files. Append the output of both of these to the accounts.ldif file via the **>>** redirection operator:

```
# ./migrate_passwd.pl /etc/passwd >> accounts.ldif
# ./migrate_group.pl /etc/group >> accounts.ldif
```

Search for the string ldapuser1 in the accounts.ldif file:

```
# grep ldapuser1 accounts.ldif
dn: uid=ldapuser1,ou=People,dc=pbm,dc=ihost,dc=com
uid: ldapuser1
cn: ldapuser1
homeDirectory: /home/ldapuser1
dn: cn=ldapuser1,ou=Group,dc=pbm,dc=ihost,dc=com
cn: ldapuser1
```

The output shows that the user was migrated to the LDIF file.

Now it is time to populate the LDAP server. This can be done while the server is not running via the **slapadd -l** command. First stop the LDAP service then add the data in the LDIF file:

```
# service ldap stop
Stopping slapd: [  OK  ]
# slapadd -l accounts.ldif
bdb_db_open: Warning - No DB_CONFIG file found in directory /var/lib/ldap: (2)
Expect poor performance for suffix dc=pbm,dc=ihost,dc=com.
```

The database is stored in the directory `/var/lib/ldap/`. For the OpenLDAP server to start, all files in this directory must be owned by the `ldap` user and group. This can be accomplished via the **`chown`** command:

```
# chown ldap:ldap /var/lib/ldap/*
```

Set the LDAP server:

```
# service ldap start
Checking configuration files for slapd:  bdb_db_open: Warning - No DB_CONFIG file found
in directory /var/lib/ldap: (2)
config file testing succeeded
[  OK  ]
Starting slapd: [  OK  ]
```

Note: Resolving the warning “**Expect poor performance for suffix dc=pbm,dc=ihost,dc=com**.” is left as an exercise for the reader :))

Test that you can query the LDAP directory for `ldapuser1` via the **`ldapsearch`** command. The **`-b`** flag sets the base of the search (it is necessary because the default suffix of `dc=pbm,dc=ihost,dc=com` has not been configured into the LDAP client; later it will not be needed):

```
# ldapsearch -x -b dc=pbm,dc=ihost,dc=com uid=ldapuser1
# extended LDIF
#
# LDAPv3
# base <dc=pbm,dc=ihost,dc=com> with scope subtree
# filter: uid=ldapuser1
# requesting: ALL
#

# ldapuser1, People, pbm.ihost.com
dn: uid=ldapuser1,ou=People,dc=pbm,dc=ihost,dc=com
uid: ldapuser1
...
# numEntries: 1
```

You should now have an OpenLDAP server installed, configured and populated with users and groups.

Deleting duplicate users

You now have a duplicate set of users and groups in both the `/etc/` file system and in LDAP. There are different ways that you can manage this duplicity. For this section, the following changes are recommended:

Delete **ldapuser1** from file system — This user was designed to be the first LDAP uid so it should not also be in the file system

Delete **root** from LDAP — Having uid of 0 (root) in LDAP is may not be a good security design.

Delete the `ldapuser1` from the file system with the **`userdel`** command, and delete `root` from LDAP via the **`ldapdelete`** command:

```
# userdel ldapuser1
# ldapdelete -x -D cn=ldaproot,dc=pbm,dc=ihost,dc=com -W \
uid=root,ou=People,dc=pbm,dc=ihost,dc=com
Enter LDAP Password: <lnx4vm>
```

Now ldapuser1’s credentials are only in LDAP and the root password is only in the local file system.

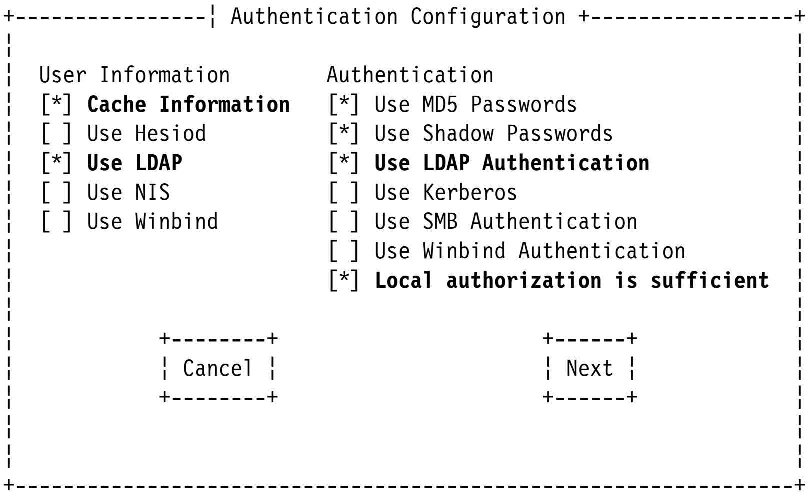
11.2.5 Configuring an LDAP client

You are now ready to configure a system to authenticate users using the new LDAP server. You will first go to a different virtual server, running on the LINUX01 user ID, and configure it to point to this LDAP server.

Start an SSH session to the Web server running on LINUX01.

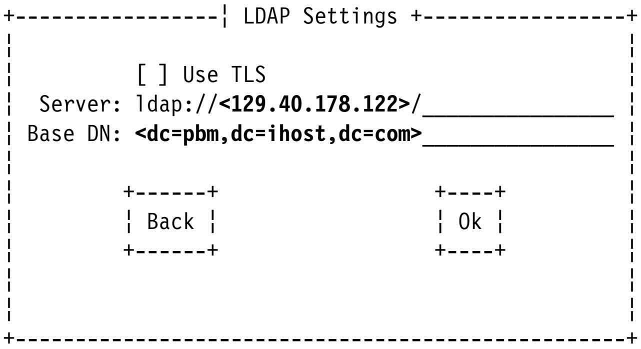
Invoke the command **authconfig-tui**. Use the **Tab** key to move between fields, the **space bar** to change selections and the **Enter** key to select:

```
# authconfig-tui
```



```
+----------------¦ Authentication Configuration +-----------------+
¦                                                                 ¦
¦  User Information        Authentication                         ¦
¦  [*] Cache Information   [*] Use MD5 Passwords                  ¦
¦  [ ] Use Hesiod          [*] Use Shadow Passwords               ¦
¦  [*] Use LDAP            [*] Use LDAP Authentication            ¦
¦  [ ] Use NIS             [ ] Use Kerberos                       ¦
¦  [ ] Use Winbind         [ ] Use SMB Authentication             ¦
¦                          [ ] Use Winbind Authentication         ¦
¦                          [*] Local authorization is sufficient  ¦
¦                                                                 ¦
¦         +--------+                       +------+               ¦
¦         ¦ Cancel ¦                       ¦ Next ¦               ¦
¦         +--------+                       +------+               ¦
¦                                                                 ¦
¦                                                                 ¦
+-----------------------------------------------------------------+
```

Under *User Information*, select **Cache Information** and **Use LDAP**. Under *Authentication*, select **Use LDAP Authentication** and **Local authorization is sufficient**. Press **Enter** when you have tabbed to the **Next** “button”.



```
+-----------------¦ LDAP Settings +-----------------+
¦                                                   ¦
¦          [ ] Use TLS                              ¦
¦  Server: ldap://<129.40.178.122>/________________ ¦
¦ Base DN: <dc=pbm,dc=ihost,dc=com>________________ ¦
¦                                                   ¦
¦        +------+                +----+             ¦
¦        ¦ Back ¦                ¦ Ok ¦             ¦
¦        +------+                +----+             ¦
¦                                                   ¦
¦                                                   ¦
+---------------------------------------------------+
```

On the next screen, set the *Server* value to point to the LDAP server. In this example, it is ldap://129.40.178.122/. Set the Base DN to your suffix value. In this example it is dc=pbm,dc=ihost,dc=com. “Press” **OK**.

Your LDAP client should now be pointing to the LDAP server. Test it via the **id ldapuser1** command:

```
# id ldapuser1
```

```
uid=500(ldapuser1) gid=500(ldapuser1) groups=500(ldapuser1)
context=root:system_r:unconfined_t:s0-s0:c0.c1023
```

Authentication should also allow LDAP. Test this by **starting another SSH session** to the LDAP server and logging in as ldapuser1:

```
login as: ldapuser1
ldapuser1@129.40.178.123's password:
```

When you get an SSH session as `ldapuser1` this shows that authentiation is working.

11.2.6 Adding new user accounts

There are different front-ends to LDAP administration, however, there is no industry standard. Details on front-end tools are beyond the scope of this section, however, it is useful to be able to add users via line commands. This section describes how to create a new LDIF file and add a new user.

Go back to the directory `/usr/share/openldap/` and view the LDIF file you created earlier, `accounts.ldif` via the **`less`** command. Search for the user `ldapuser1` via the **`/`** subcommand:

```
# cd /usr/share/openldap/migration
# less accounts.ldif
/ldapuser1
```

Before:

```
dn: uid=ldapuser1,ou=People,dc=pbm,dc=ihost,dc=com
uid: ldapuser1
cn: ldapuser1
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 13515
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 500
gidNumber: 500
homeDirectory: /home/ldapuser1
```

Copy and paste the lines for that user to a new file `ldapuser.ldif`. Change the name `ldapuser1` to the desired new name (`ldapuser2` in this example). Also increment the uidNumber field (to `501` in this example)

```
# vi ldapuser.ldif
```

After:

```
dn: uid=ldapuser2,ou=People,dc=pbm,dc=ihost,dc=com
uid: ldapuser2
cn: ldapuser2
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 13515
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 501
```

```
gidNumber: 500
homeDirectory: /home/ldapuser2
```

You now have created a new LDIF file containing the entries for a new LDAP user with a unique UID. Now you can add the user to the LDAP database via the following **ldapadd** command:

```
# ldapadd -x -D cn=ldaproot,dc=pbm,dc=ihost,dc=com -W -f ldapuser.ldif
Enter LDAP Password:

adding new entry "cn=ldapuser2,ou=Group,dc=pbm,dc=ihost,dc=com"
```

Set a password

You can set the new user's password via the following **ldappasswd** command:

```
# ldappasswd -x -D cn=ldaproot,dc=pbm,dc=ihost,dc=com -W -s lnx4vm \
uid=ldapuser2,ou=People,dc=pbm,dc=ihost,dc=com
Enter LDAP Password: <lnx4vm>

Result: Success (0)
```

You can test logging into a virtual server with the new user's credentials. If you successfully configured the virtual server running on LINUX01 to point to the LDAP server, try a new SSH session with the new user. You should be able to successfully login:

```
login as: ldapuser2
ldapuser2@129.40.178.182's password:
Could not chdir to home directory /home/ldapuser2: No such file or directory
```

The issue of not having a home directory will be addressed in section 12.3, "Centralizing home directories for LDAP users" on page 189.

11.3 Creating a virtual file and print server

Samba allows Windows clients to map Linux file systems as shared drives. Samba can also act as a middle-man between Windows clients and a Linux print server. The recommended Linux print server is CUPS - the Common UNIX Printing System. This section does not describe the configuration of CUPS but it does describe how the necessary RPMs are installed.

The steps in this section are as follow:

- "Cloning a Linux virtual server" on page 173
- "Installing necessary RPMs" on page 174
- "Configuring Samba configuration file" on page 175
- "Adding a Samba user" on page 175
- "Starting Samba at boot time" on page 176
- "Testing your changes" on page 176

11.3.1 Cloning a Linux virtual server

From the controller, clone a basic virtual server. In this example the user ID LINUX03 is used.

```
# clone.sh linux03
Invoking CP command: QUERY LINUX03
HCPCQU045E LINUX03 not logged on
Error: non-zero CP response for command 'QUERY LINUX03': #45
```

```
Setting device 0.0.0191 offline
Done
Setting device 0.0.0191 online
Done
LINUX03  CONF-RH5 D1 F         80         13          1 12/22/2006 13:20:03

WARNING!!: this will copy disk 100 to LINUX03 100
New host name will be:  lat183.pbm.ihost.com
New TCP/IP address will be: 129.40.178.123
Other network data is retrieved from LINUX03 CONF-RH5 on 191 disk
Are you sure you want to overwrite these disks (y/n): y
Please enter a brief description of this server
This is a Samba server running RHEL5
...
Successfully cloned /rhel5master to LINUX03
You should be able to ping 129.40.178.283 within one minute
```

Start an SSH session to the new virtual server.

11.3.2 Installing necessary RPMs

Add the following RPMs via the **yum -y** command:

```
# yum -y install samba-common samba samba-client
...
=============================================================================
 Package                 Arch       Version          Repository        Size
=============================================================================
Installing:
 samba                   s390x      3.0.23c-2        RHEL5              16 M
 samba-client            s390x      3.0.23c-2        RHEL5             4.6 M
 samba-common            s390       3.0.23c-2        RHEL5             9.2 M
 samba-common            s390x      3.0.23c-2        RHEL5             9.1 M
Installing for dependencies:
 popt                    s390       1.10.2-36.el5    RHEL5              69 k

Transaction Summary
=============================================================================
Install      5 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 39 M
Downloading Packages:
Running Transaction Test
warning: samba-client-3.0.23c-2: Header V3 DSA signature: NOKEY, key ID 897da07a
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: samba-common                 ######################### [1/5]
  Installing: popt                         ######################### [2/5]
  Installing: samba-client                 ######################### [3/5]
  Installing: samba                        ######################### [4/5]
  Installing: samba-common                 ######################### [5/5]

Installed: samba.s390x 0:3.0.23c-2 samba-client.s390x 0:3.0.23c-2 samba-common.s390
0:3.0.23c-2 samba-common.s390x 0:3.0.23c-2
Dependency Installed: popt.s390 0:1.10.2-36.el5
Complete!
```

Confirm that the RPMs were added:

```
# rpm -qa | grep samba
samba-common-3.0.23c-2
samba-common-3.0.23c-2
samba-3.0.23c-2
samba-client-3.0.23c-2
```

11.3.3 Configuring Samba configuration file

The one configuration file for Samba is /etc/samba/smb.conf. It is easy to add an SMB share that will be made available by the Samba server. A good test directory is /usr/share/doc/ as it has much good Linux documentation. The following example will create a file *share* named sharedoc:

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf    // add three lines at the bottom of the file:
...
[sharedoc]
  comment = RHEL 5 on System z documentation
  path = /usr/share/doc/
```

You can verify the syntax of your changes with the testparm command:

```
# testparm smb.conf
Load smb config files from smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[sharedoc]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
...
```

This change will create an SMB share named **sharedoc** consisting of the contents of the directory /usr/share/doc and below.

11.3.4 Adding a Samba user

The default method that Samba uses to determines users' credentials is to look in the /etc/samba/smbpasswd file. That user must first exist in the Linux file system (/etc/passwd, /etc/shadow, etc). To create a new Samba user, the **smbpasswd -a** command is used. First use the **useradd** and **passwd** commands to add a user locally. In this example, the user sambauser1 is used:

```
# id sambauser1
id: sambauser1: No such user
# useradd sambauser1
# passwd sambauser1
Changing password for sambauser1.
New password: <lnx4vm>
Re-enter new password: <lnx4vm>
Password changed
```

Add the user sambauser1 to the smbpasswd file via the **smbpasswd -a** command:

```
# smbpasswd -a sambauser1
New SMB password: <lnx4vm>
Retype new SMB password: <lnx4vm>
```

```
startsmbfilepwent_internal: file /etc/samba/smbpasswd did not exist. File successfully
created.
account_policy_get: tdb_fetch_uint32 failed for field 1 (min passwd length), returning 0
...
Added user sambauser1.
```

You can see that the last **smbpasswd** command added `sambauser1` to the file `smbpasswd`:

```
# cat smbpasswd
sambauser1:502:2E6F0C45D305054CAAD3B435B51404EE:74154D10CC18CFC98F1ED128DB8764DA:[U
]:LCT-45A3AAB2:
```

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 a simple as pointing the virtual file and print server at the virtual LDAP server as described in “Creating a virtual LDAP server” on page 165 because the Samba schema must first be added to LDAP. Details are outside the scope of this book.

11.3.5 Starting Samba at boot time

Samba can be started for the current session via the **service** command and at boot time via the **chkconfig** command:

```
# service smb start
Starting SMB services: [  OK  ]
Starting NMB services: [  OK  ]
# chkconfig smb on
```

Samba should now be running and configured to start at boot time.

11.3.6 Testing your changes

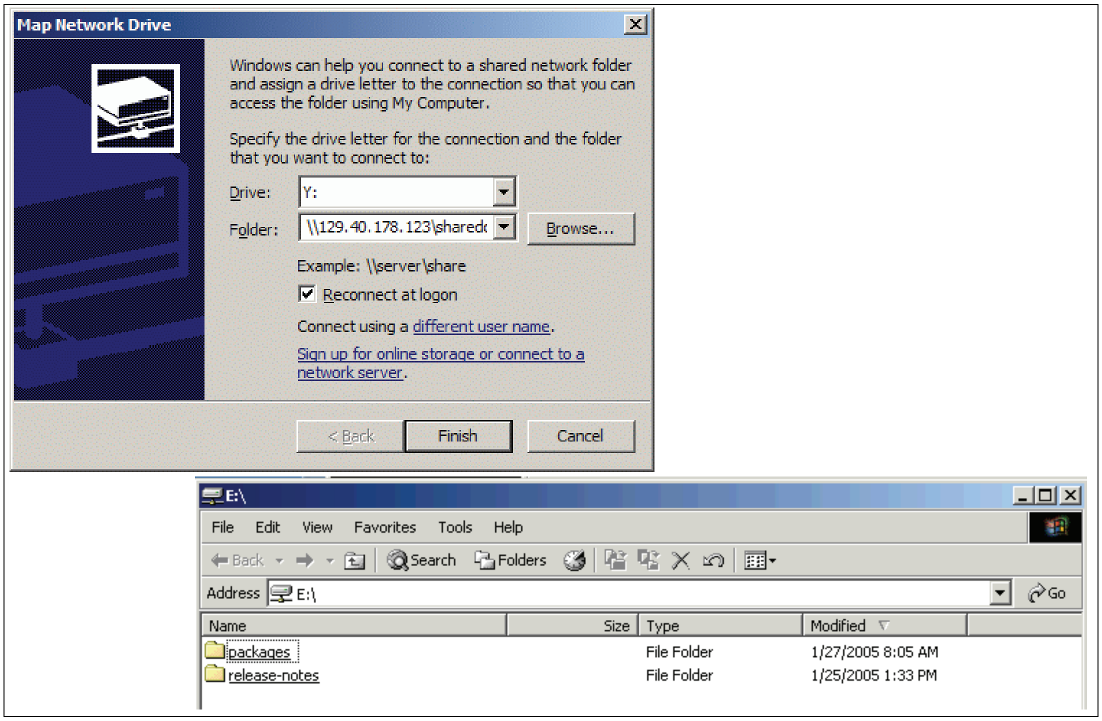
You can verify that Samba is running via the following **service** command:

```
# service smb status
smbd (pid 6987 6982) is running...
nmbd (pid 6985) is running...
```

You can verify the shares that are available via the following **smbclient** command:

```
# smbclient -U sambauser1 -L localhost
Password:

Domain=[LAT123] OS=[Unix] Server=[Samba 3.0.23c-2]

        Sharename       Type      Comment
        ---------       ----      -------
        sharedoc        Disk      RHEL 5 on System z documentation
        IPC$            IPC       IPC Service (Samba Server)
        sambauser1      Disk      Home Directories
Domain=[LAT123] OS=[Unix] Server=[Samba 3.0.23c-2]
...
```

You can test getting a Samba share from a Windows desktop. Go to any Windows Explorer window (such as *My Computer*) and select **Tools** -> **Map Network Drive**. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 11-2 on page 177. In this example the UNC is **\\129.40.178.123\sharedoc**. Then click **Finish**. If all the steps were correct, you should see the files in a new Explorer window as shown in the bottom right corner of the figure.



Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Y:

Folder: \\129.40.178.123\sharedc

Example: \\server\share

☑ Reconnect at logon

Connect using a different user name.

Sign up for online storage or connect to a network server.

Address E:\

Name	Size	Type	Modified
packages		File Folder	1/27/2005 8:05 AM
release-notes		File Folder	1/25/2005 1:33 PM

Figure 11-2 Mapping a network drive to the Samba server

You should now have Samba configured and running with one new share available.

If you prefer a DOS command line, you can also link to the share via the following **net use** command:

```
c:\>net use y: \\129.40.178.123\sharedoc
The password is invalid for \\129.40.178.123\sharedoc.

Enter the user name for '129.40.178.123': sambauser1
Enter the password for 129.40.178.123:
The command completed successfully.
```

You can detach the share via the following **net use** command:

```
c:\>net use y: /delete
y: was deleted successfully.
```

11.3.7 Configuring printing

Configuring printing is more complex and is beyond the scope of this section. For details see the Redpaper *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864, on the Web at:

http://www.redbooks.ibm.com/abstracts/redp3864.html

11.4 Creating a virtual application development server

Most Linux distributions come with a basic set of application development tools, making Linux one of the most versatile development systems. These basic tools are ideal for projects of any size.

The development languages used in implementation range from scripting languages such as Python or Tcl, to compiled languages such as C/C++ and Java. There are software available on Linux to help form a development system for developers to create integrated applications. MySQL and Apache are among them. A popular open source Web platform is LAMP, which stands for the open source software and programming languages used to make up the platform: Linux, Apache, MySQL, Python or PHP.

From the controller clone a virtual server to the user ID `LINUX04`.

```
# clone.sh linux04
...
```

Note that before installing the development tools, the root file system is about half full:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  1.1G  1.1G  49% /
tmpfs                 124M     0  124M   0% /dev/shm
```

SSH in to the new virtual server. You can use the **`yum -y groupinstall`** command to install the groups named **development-tools** and **development-libs**. This will add about 181 packages which requires a number of minutes to complete:

```
# yum -y groupinstall development-tools development-libs
...
Transaction Summary
=============================================================================
Install     181 Package(s)
Update        0 Package(s)
Remove        0 Package(s)
...
Complete!
```

Your application development server is now ready to use. You may choose to add or remove different packages. Use **`df -h`** command to show that your system should still have about 30% free space:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  1.5G  635M  71% /
tmpfs                 124M     0  124M   0% /dev/shm
```

11.4.1 Additional resources

The following Web sites are resources for additional information on application development topics:

Scripting languages

http://www.perl.com/
http://www.python.org/
http://www.freeos.com/guides/lsst/

C/C++

http://gcc.gnu.org/onlinedocs/gcc/
http://en.wikipedia.org/wiki/GNU_Compiler_Collection#External_links
http://vertigo.hsrl.rutgers.edu/ug/make_help.htmll
http://www.gnu.org/software/make/manual/html_chapter/make_toc.html

Java

http://www-130.ibm.com/developerworks/java/

http://java.sun.com/
http://csdl.ics.hawaii.edu/~johnson/613f99/modules/04/jar-files.html
http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html

Linux kernel development

http://www.kernel.org/pub/linux/docs/lkml/#blkd

Web development

http://www.onlamp.com/
http://cgi.resourceindex.com/
http://www.perl.com/

12

Miscellaneous Recipes

Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.

--Albert Einstein

This chapter has the following sections of miscellaneous tasks that you might want to perform:

12.1 Adding a logical volume

There are times when you require more disk space than a single direct access storage device (DASD) volume provides. For example, if you want to have a shared `/home/` directory you will want it to be of sufficient size. When this is the case, you can use the Logical Volume Manager (LVM) to combine multiple DASD volumes into one logical volume.

The following process describes how to create a logical volume, or extend an existing logical volume with additional DASD on a Linux guest. The overall steps in creating a logical volume are as follows:

- Add minidisks to the z/VM directory entry and IPL Linux
- Bring the new DASD online
- Format and partition the DASD
- Create the logical volume and file system
- Update the file system table
- Make the change persistent

12.1.1 Adding minidisks to the z/VM directory entry

The overall steps are the following. Specific details of these steps on z/VM are not given.

- Determine the labels of the volumes that will be added.

- Add minidisk statements to define minidisks (at virtual addresses 103 and 104 of size 6678 cylinders in this example) to the appropriate Linux user ID definition in the `USER DIRECT` file.
- Create the `USER DISKMAP` file to verify the disk layout
- Bring the changes online with the **`DIRECTXA`** command
- Modify the Linux system to recognize the new minidisks
- Shutdown the system and logoff the z/VM user ID. **The LINUX02 user ID is used** in the following example (this puts the logical volume on the LDAP server)
- Logon to the user ID and verify the new minidisks are available.

12.1.2 Making the new minidisks available

Before you shut down, make a backup copy then modify the `/etc/modprobe.conf` file to include the new minidisks at virtual addresses 103 and 104:

```
# cd /etc
# cp modprobe.conf modprobe.conf.orig
# vi modprobe.conf
alias eth0 qeth
options dasd_mod dasd=100-104
```

Save the file. To effect the change, you must build a new initial RAMdisk with the following **`mkinitrd`** command, then run **`zipl`** to write the changes to the boot record:

```
# mkinitrd -v -f /boot/initrd-$(uname -r).img $(uname -r)
Creating initramfs
...
Adding module jbd
Adding module ext3
Adding module dasd_mod with options dasd=100-104
Adding module dasd_eckd_mod
Adding module dasd_fba_mod
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux' (default)
Preparing boot device: dasda (0100).
Done.
# shutdown -h now
...
```

When your system comes back up, **start an SSH session** to it. Use the **`lsdasd`** command to verify that the new minidisks have been recognzied

```
# lsdasd
0.0.0100(ECKD) at ( 94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at ( 94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at ( 94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0103(ECKD) at ( 94:12) is dasdd : active at blocksize 4096, 1202040 blocks, 4695 MB
0.0.0104(ECKD) at ( 94:16) is dasde : active at blocksize 4096, 1202040 blocks, 4695 MB
```

12.1.3 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 in parallel:

```
# for i in d e
```

```
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 2713
[2] 2714
```

When the jobs are finished use the **fdasd** command with the **-a** flag to create a single partition from each minidisk:

```
# fdasd -a /dev/dasdd
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
# fdasd -a /dev/dasde
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The minidisks are now ready for you to use in the logical volume. If you are creating a new logical volume, see 12.1.4, “Create the logical volume and file system” on page 184. If you are extending an existing logical volume, skip ahead to 12.2, “Extending an existing logical volume” on page 187.

12.1.4 Create the logical volume and file system

The overall steps involved in creating a logical volume are:

- Create physical volumes from the two DASD
- Create a single volume group
- Create a single logical volume
- Make a file system from the logical volume

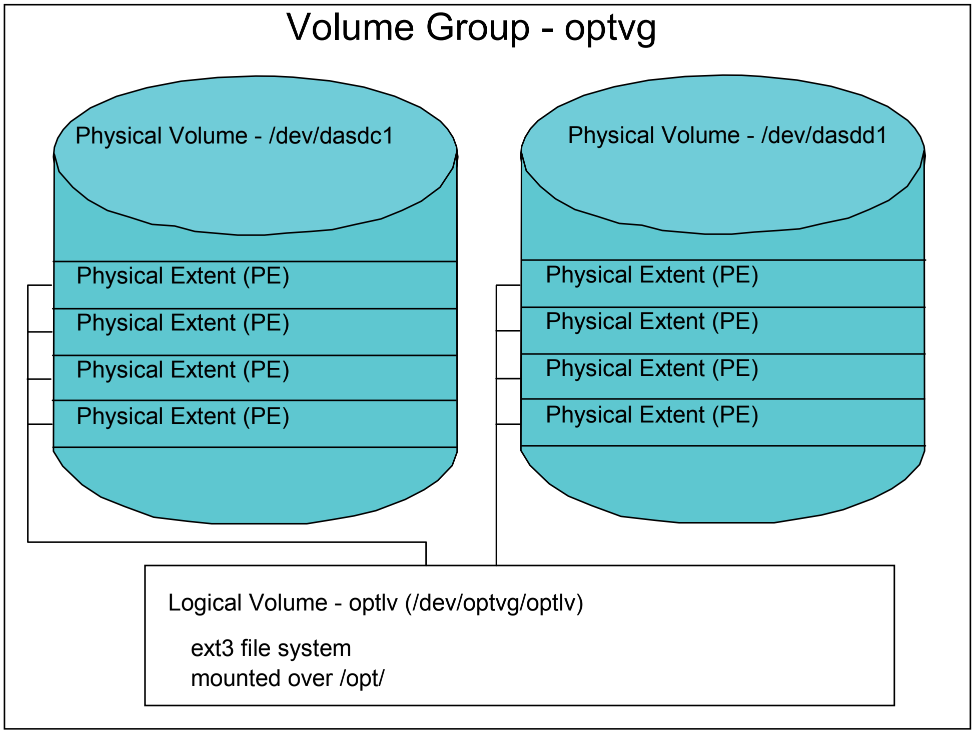


Figure 12-1 LVM block diagram

Figure 12-1 shows a block diagram of the logical volume manager reflecting this example.

Creating physical volumes from the two DASD

The **pvcreate** command initializes DASD for use by LVM. Initialize the two new DASD partitions. Verify with the **pvdisplay** command:

```
# pvcreate /dev/dasdd1 /dev/dasde1
  Physical volume "/dev/dasdd1" successfully created
  Physical volume "/dev/dasde1" successfully created

# pvdisplay
--- NEW Physical volume ---
  PV Name               /dev/dasdd1
  VG Name
  PV Size               4.59 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               eksaUw-HvSX-S9m8-a5u0-RBOn-fMH7-y6fkRb

  --- NEW Physical volume ---
  PV Name               /dev/dasde1
  VG Name
  PV Size               4.59 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               y6lMf2-PkCk-w46g-lSkl-3M9d-7vUZ-rtvaJX
```

Creating a single volume group

The **vgcreate** command can be used to create a volume group named homevg from the two DASD. Use the **vgdisplay** command to verify:

```
# vgcreate homevg /dev/dasdd1 /dev/dasde1

# vgdisplay
  --- 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               9.16 GB
  PE Size               4.00 MB
  Total PE              2346
  Alloc PE / Size       0 / 0
  Free  PE / Size       2346 / 9.16 GB
  VG UUID               IHAq2g-ME3w-b6LY-E3hA-raNk-bFOG-Poprma
```

In this example, there are 2346 free physical extents.

Creating a single logical volume

The **lvcreate** command is used to create a logical volume. The **-l 2346** flag specifies to use all free extents, 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. Use the **lvdisplay** command to verify:

```
# lvcreate -l 2346 -n homelv homevg
  Logical volume "homelv" created
# lvdisplay
--- Logical volume ---
  LV Name                /dev/homevg/homelv
  VG Name                homevg
  LV UUID                roBwTM-sxkF-AdLn-sePN-KnzT-n36r-7lkIxk
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                9.16 GB
  Current LE             2346
  Segments               2
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0
```

Making a file system from the logical volume

Now you have a logical volume. 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 1.36 (05-Feb-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
402400 inodes, 803840 blocks
40192 blocks (5.00%) reserved for the super user
...
```

The file system created from the logical volume is now ready to be mounted.

12.1.5 Updating the file system table

You could now mount the file system manually. However if you add the mount to the file system table file, `/etc/fstab`, you can effectively test the change by using the **`mount`** command with only one argument. Add one line to the file:

```
# cd /etc
# vi fstab
LABEL=/              /                        ext3    defaults        1 1
/dev/dasdb1          swap                     swap    defaults        0 0
/dev/dasdc1          swap                     swap    defaults        0 0
/dev/homevg/homelv   /home                    ext3    defaults        0 0
...
```

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. Use the **`df -h`** command to verify that it is mounted.

```
# mount /home
# ls /home
lost+found/
# df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/dasda1          2.3G  1.2G 1016M  53% /
tmpfs                124M     0  124M   0% /dev/shm
/dev/mapper/homevg-homelv
                     9.1G  149M  8.5G   2% /home
```

Moving data from existing file system

You may have noticed that there may be some data in the directory which will serve as the mount point. To complete the addition of the logical volume, it is recommended that you move any data from the existing directory to the new logical volume. First unmount the logical volume and view the contents of the `/home/` directory:

```
# umount /home
# ls -F /home
ldapuser1/
```

In this example, there is only one subdirectory. You can move the contents from the root file system to the new logical volume a number of different ways. Following is one method:

```
# cd /home
# mkdir /tmp/home
# mv * /tmp/home
# cd ..
# mount /home
# cd /home
# mv /tmp/home/* .
# ls -F
ldapuser1/  lost+found/
```

Even though you tested mounting the file system via reading the /etc/fstab file, you may want to test a reboot to verify the logical volume is successfully mounted.

12.2 Extending an existing logical volume

This section describes the process of adding a new minidisk to an existing LVM. This is useful when your logical volume has run out of space. In this example, the **vgdisplay** command shows that the existing volume group is full.

```
# vgdisplay
--- Volume group ---
  VG Name               homevg
...
  Total PE              2346
  Alloc PE / Size       2346 / 9.16 GB
  Free  PE / Size       0 / 0
  VG UUID               IHAq2g-ME3w-b6LY-E3hA-raNk-bFOG-Poprma
```

First, repeat the steps at the beginning of this section to add a new minidisk. In this example, a minidisk at virtual address 105 is added of size 3338 cylinders.

Create a new RAMdisk to enable the new minidisk

Edit the /etc/modprobe.conf file and add the new minidisk at address 105 to the existing range:

```
# vi /etc/modprobe.conf
alias eth0 qeth
options dasd_mod dasd=100-105
```

Use the **mkinitrd** command to rebuild the initial RAMdisk, then the **zipl** command to rewrite the boot sector on the root partition.

```
# mkinitrd -v -f /boot/initrd-$(uname -r).img $(uname -r)
...
# zipl
...
```

Reboot the system to test the changes.

```
# reboot
...
```

When your system comes back, verify that the new minidisk is available via the **lsdasd** command:

```
# lsdasd
...
0.0.0100(ECKD) at ( 94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at ( 94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at ( 94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
```

```
0.0.0103(ECKD) at ( 94:12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at ( 94:16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0105(ECKD) at ( 94:20) is dasdf : n/f
```

Format the minidisk via the **dasdfmt** command and make a single partition via the **fdasd -a** command as you did before:

```
# dasdfmt -b 4096 -y -f /dev/dasdf
Finished formatting the device.
Rereading the partition table... ok
# fdasd -a /dev/dasdf
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/dasdf1
  Physical volume "/dev/dasdf1" successfully created
```

Extending the volume group

Use the **vgextend** command to extend the volume group into the new physical volume. Then, use **vgdisplay** to verify that the volume group has free space.

```
# vgextend homevg /dev/dasdf1
  Volume group "homevg" successfully extended
# vgdisplay
--- Volume group ---
  VG Name               homevg
  ...
  Total PE              2932
  Alloc PE / Size       2346 / 9.16 GB
  Free  PE / Size       586 / 2.29 GB
  VG UUID               IHAq2g-ME3w-b6LY-E3hA-raNk-bFOG-Poprma
```

Note there are 586 new free physical extents (PEs) in this example.

Extend the logical volume and the file system

Now that you have free space in the volume group, you can increase the size of the existing logical volume with the **lvextend** command. The **-l** option specifies the number extents to add. Finally, use the **resize2fs** command, which increases the size of the file system while it is 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 | grep home
/dev/mapper/homevg-homelv
                      9.1G  149M  8.5G   2% /home
# lvextend -l +586 /dev/homevg/homelv
  Extending logical volume homelv to 11.45 GB
  Logical volume homelv successfully resized
# resize2fs -p /dev/homevg/homelv
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/homevg/homelv is mounted on /home; on-line resizing required
Performing an on-line resize of /dev/homevg/homelv to 3002368 (4k) blocks.
The filesystem on /dev/homevg/homelv is now 3002368 blocks long.

# df -h | grep home
/dev/mapper/homevg-homelv
                       12G  152M   11G   2% /home
```

You have now added the new minidisk as free space to the existing logical volume. Test your changes by rebooting your system:

```
# reboot
...
```

12.3 Centralizing home directories for LDAP users

If you have completed “Creating a virtual LDAP server” on page 165, you have a working LDAP authentication server that provides centralized login authentication. If you have completed section 12.1, “Adding a logical volume” on page 181, you have a larger file system mounted over `/home/` for storing user data. Now you can bring these together with the automount service to have a centralized /home file system. In this fashion, users can login to all virtual servers using the same credentials and being able to access the same data. Further, you can modify the golden image so that all newly cloned virtual servers will inherit this setup. Following is a block diagram

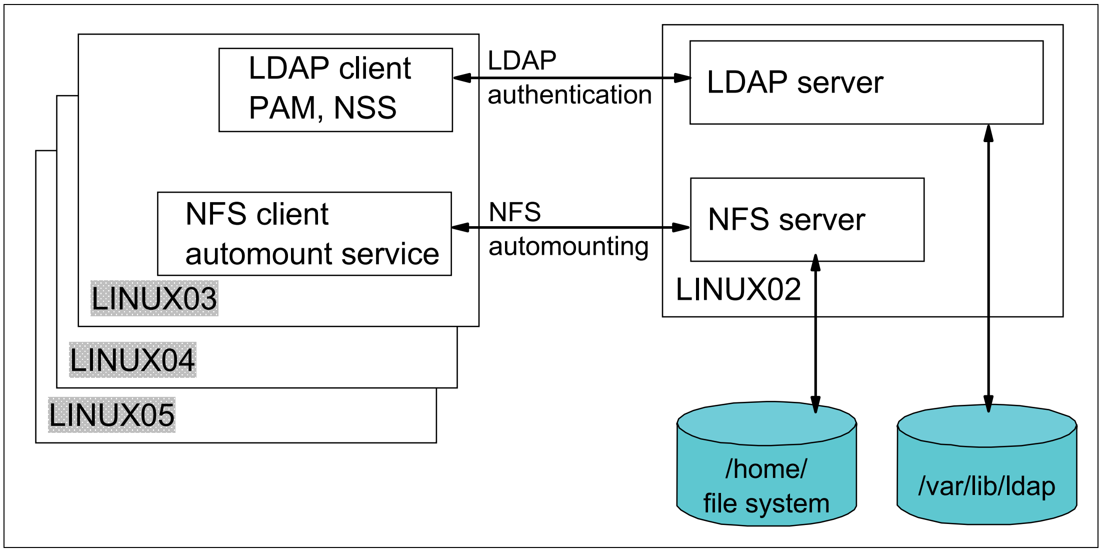


Figure 12-2 Block diagram of centralized LDAP authentication and automounted /home file system

12.3.1 Configuring the NFS server

Configure `LINUX02` to export `/home` over NFS. This is similar to the configuration of the controller in “Turning on the NFS server” on page 125.

Start an SSH session to the virtual server running on `LINUX02`. Add one line to the `/etc/exports` file:

```
# vi /etc/exports
/home           *(rw,sync)
```

Set the NFS service to start at boot time via the **`chkconfig`** command and for this session via the **`service`** command:

```
# chkconfig nfs on
# service nfs start
Starting NFS services:  [  OK  ]
Starting NFS quotas: [  OK  ]
Starting NFS daemon: [  OK  ]
Starting NFS mountd: [  OK  ]
```

Test mounting the newly exported file system locally:

```
# mount localhost:/home /mnt
# ls /mnt
ldapuser1  lost+found
```

You now have /home/ available for NFS mounting.

12.3.2 Configuring the master image for LDAP and automount

In this section you will shut down the controller and boot the master image. You will then configure it for LDAP authentication and automount of `/home/` to the virtual server running on `LINUX02`.

Start a 3270 connect to the controller, shut it down, and IPL the master image:

```
Red Hat Enterprise Linux Server release 4.92 (Tikanga)
Kernel 2.6.18-1.2839.el5 on an s390x

lat128 login: root
Password:

Last login: Fri Dec 15 09:49:10 from 9.56.60.150
#shutdown -h now
...
==> ipl 100
...
==> #cp disc
```

Configuring LDAP

You can modify the authentication settings via the command **`authconfig-tui`** (you can also use **`authconfig`** from a graphical environment such as VNC).

Use the **Tab** key to move between fields, the **space bar** to change selections and the **Enter** key to select:

```
# authconfig-tui

      +----------------¦ Authentication Configuration +-----------------+
      ¦                                                                 ¦
      ¦  User Information       Authentication                          ¦
      ¦  [*] Cache Information  [*] Use MD5 Passwords                   ¦
      ¦  [ ] Use Hesiod         [*] Use Shadow Passwords                ¦
      ¦  [*] Use LDAP           [*] Use LDAP Authentication             ¦
      ¦  [ ] Use NIS            [ ] Use Kerberos                        ¦
      ¦  [ ] Use Winbind        [ ] Use SMB Authentication              ¦
      ¦                         [ ] Use Winbind Authentication          ¦
      ¦                         [*] Local authorization is sufficient   ¦
      ¦                                                                 ¦
      ¦          +--------+                      +------+               ¦
      ¦          ¦ Cancel ¦                      ¦ Next ¦               ¦
      ¦          +--------+                      +------+               ¦
      ¦                                                                 ¦
      ¦                                                                 ¦
      +-----------------------------------------------------------------+
```

Under *User Information*, select **Cache Information** and **Use LDAP**. Under *Authentication*, select **Use LDAP Authentication** and **Local authorization is sufficient**. Press **Enter** when you have tabbed to the **Next** “button”.

```
+-----------------¦ LDAP Settings +-----------------+
¦                                                   ¦
¦          [ ] Use TLS                              ¦
¦  Server: ldap://<129.40.178.122>/_______________  ¦
¦ Base DN: <dc=pbm,dc=ihost,dc=com>_______________  ¦
¦                                                   ¦
¦         +------+                +----+            ¦
¦         ¦ Back ¦                ¦ Ok ¦            ¦
¦         +------+                +----+            ¦
¦                                                   ¦
¦                                                   ¦
¦                                                   ¦
+---------------------------------------------------+
```

Set the *Server* value to point to the LDAP server. In this example, it is `ldap://129.40.178.122/`. Set the `Base DN` to your suffix value. In this example it is `dc=pbm,dc=ihost,dc=com`. “Press” **OK**.

Your LDAP client should now be pointing to the LDAP server. Test it via the **`id ldapuser1`** command:

```
# id ldapuser1
uid=500(ldapuser1) gid=500(ldapuser1) groups=500(ldapuser1)
context=root:system_r:unconfined_t:s0-s0:c0.c1023
```

Authentication should also allow LDAP. Test this by **starting another SSH session** to the LDAP server and logging in as ldapuser1:

```
login as: ldapuser1
ldapuser1@129.40.178.122's password:
```

When you get an SSH session as `ldapuser1` this shows that authentiation is working.

Configuring automount

Now SSH to the master image to configure the automounter. Add an entry to the bottom of `/etc/auto.master` for `/home`.

```
# vi /etc/auto.master
/nfs    /etc/auto.controller
/home   /etc/auto.home
```

Create the file `/etc/auto.home` and add one line to it:

```
# vi /etc/auto.home
*       <129.40.178.122>:/home/&
```

The asterisk (*) is used as a wildcard character to represent the local subdirectory. The ampersand (&) is used as the remote system name or the remote subdirectory. Whatever is in the local directory name field replaces the ampersand character.

Finally, restart **`autofs`** and test logging in as `ldapuser1` with **`su`**. You will see that /home/ldapuser1 is now mounted from the LDAP server:

```
# service autofs restart
Stopping automount: [  OK  ]
Starting automount: [  OK  ]
# su - ldapuser1
$ pwd
/home/ldapuser1
$ mount | grep ldapuser1
129.40.178.122:/home/ldapuser1 on /home/ldapuser1 type nfs (rw,addr=129.40.178.122)
```

You have now configured the automounter to mount a remote LDAP user's home directory at login.

You can now shut down the master image, and IPL the controller (200). Clone a new virtual server via the **clone.sh** script. Start an SSH session to the new virtual server and LDAP authentication and the automounting of /home/ via NFS should work.

12.4 Rescuing a Linux system

This section describes how to boot your Linux server into different modes for troubleshooting purposes. It covers booting Linux into single user mode, and also entering a rescue environment when you require more advanced troubleshooting.

12.4.1 Entering single user mode

Single user mode is helpful when you need to recover the root password, or if you are having problems while booting Linux into the default runlevel. To enter single user mode, first IPL your Linux server from the 3270 console. You will see a message similar to:

```
zIPL v1.3.2 interactive boot menu
0. default (linux)
1. linux
Note: VM users please use '#cp vi vmsg <input>'
Please choose (default will boot in 15 seconds):
```

Use the **#cp vi vmsg** command to boot the desired menu option (zero in this example), followed by the number one for single user mode:

```
==> #cp vi vmsg 0 1
```

After some initial kernel boot messages, you should see:

```
INIT: Entering runlevel: 1
Entering non-interactive startup
Telling INIT to go to single user mode.
INIT: Going single user
INIT: Sending processes the TERM signal
INIT: Sending processes the KILL signal
sh-3.00#
```

In single user mode, you are logged in as the root user. You can use the **passwd** command to set the root password. All of the file systems in /etc/fstab are mounted, but networking has not been started. To exit single user mode, you can type **reboot**, or enter **init 3** to continue booting normally.

12.4.2 Entering a rescue environment

If you encounter errors mounting the root file system, or have other problems that prevent you from entering single user mode, you can enter a rescue environment. This environment loads a Linux image in memory, and does not attempt to mount the root file system.

To enter a rescue environment, initiate an interactive Linux installation. For more detailed information about interactive installation, see Chapter 7., "Installing and configuring RHEL 5" on page 89. Perform the following steps:

1. Logon to LNXMAINT. Copy the RHEL5 EXEC file to a new file named RESCUE EXEC, and copy the user's PARM-RH5 file to a new file (LINUX07 RESCUE in this example):

```
==> copy rhel5 exec d rescue = =
==> copy linux07 parm-rh5 d = rescue =
```

2. Next, edit `RESCUE EXEC` to point to the new `LINUX07 RESCUE` file:

```
==> xedit RESCUE EXEC
...
00005 'PUN RHEL5        KERNEL * (NOH'
00006 'PUN' userid() '  RESCUE * (NOH'
00007 'PUN RHEL5        INITRD * (NOH'
...
==> file
```

3. Edit the `LINUX07 RESCUE` file, replacing any kickstart or VNC lines with the **rescue** command line option:

```
==> xedit  LINUX07 RESCUE
ramdisk_size=40000 root=/dev/ram0 ro ip=off
CMSDASD=191 CMSCONFFILE= LINUX07.CONF-RH5
rescue
```

4. Now **logoff of `LNXMAINT`** and **logon to `LINUX07`.** Answer **no** to IPL from DASD question, and run `RESCUE EXEC`.

```
==> RESCUE
Linux version 2.6.18-1.2839.el5 (brewbuilder@devel8.z900.redhat.com) (gcc versio
n 4.1.1 20061130 (Red Hat 4.1.1-43)) #1 SMP Tue Dec 5 22:58:42 EST 2006
We are running under VM (64 bit mode)
Detected 2 CPU's
Boot cpu address  0
Built 1 zonelists.  Total pages: 65536
Kernel command line: ramdisk_size=40000 root=/dev/ram0 ro ip=off
                     CMSDASD=191 CMSCONFFILE= LINUX07.CONF-RH5
                     rescue
...
```

The install process directs you to telnet or SSH to the IP address of your Linux server to begin the first stage of the installation.

5. Use SSH to connect to the IP address and log in as root. After choosing the language, the rescue environment will prompt you for the location of the rescue image, which is located in the install tree on the controller.

6. Press **F2** to allow for additional devices. Choose **NFS image**, then enter the IP address of the controller and the path `/nfs/rhel5`.

7. The rescue image will search for your Linux installation and will prompt you to mount the partitions it finds. After this, you will be presented with a shell prompt. If you chose to mount your root partition, it will be mounted to the `/mnt/sysimage` directory.

   Note: if the rescue image cannot find your partition, you can try to mount it yourself via the **`mount`** command. For example:

```
# mount /dev/dasda1 /mnt/runtime/
# ls /mnt/runtime/
bin         home        media       root        sys
boot        lib         mnt         sbin        tmp
dev         lib64       opt         selinux     usr
etc         lost+found  proc        srv         var
```

8. Type **`exit`** to leave the shell and exit rescue mode.

Part 3

Common tasks

This part of the book discusses tasks common to z/VM, SLES 10 and RHEL 5.

13

Monitoring z/VM and Linux

> *Not everything that can be counted counts, and not everything that counts can be counted*.
>
> --Albert Einstein

This chapter briefly describes how to monitor z/VM and Linux. For a more thorough chapter on z/VM performance and monitoring, see Chapter 11, *Monitoring performance and capacity*, in the Manual *Getting Started With Linux*, SC24-6096 on the Web at:

`http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/hcsh2a70`

There are a number of z/VM monitoring tools such as Computer Associates' VM:Monitor, IBM's z/VM Performance Toolkit and Velocity Software's ESALPS. The IBM z/VM Performance Toolkit is briefly described in this section.

For more information on Computer Associates' VM:Monitor, see:

`http://www.ca.com/`

For more information on Velocity's ESALPS, see:

`http://www.velocitysoftware.com/`

This chapter describes the following topics:

- "Using INDICATE and other commands" on page 197
- "The z/VM Performance Toolkit" on page 201
- "Monitoring Linux" on page 208
- "Registering Linux images with the Performance Toolkit" on page 210

13.1 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.

13.1.1 Using the INDICATE command

z/VM has some basic commands such as **INDICATE**. There are many parameters that can be included. Use the command **HELP INDICATE** for a basic understanding and then press **F11** for help on each parameter.

INIDICATE LOAD

If no parameter is specified **INDICATE LOAD** is the default option. There are two flavors of this, a class G and a class E. Class G users can use **INDICATE** to display recent contention for system resources, display environment characteristics and measurements of resources used by their virtual machine.

The output from user ID with class E privilege (e.g. MAINT, OPERATOR) is shown here. The lines are number 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 where z/VM performance analysts tend to focus are the following:

- AVGPROC on line 1 gives the overall processor utilization, 38% in this example. The number following it is the number of 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 ready to run virtual processors to keep the physical processors busy. Line 2 describes paging to expanded storage. Most z/VM systems on z900 class machines can sustain 1000s of paging operations a second without any problems. The MIGRATE rate is the number of pages per second being moved from expanded storage out to paging space on DASD. A healthy system will have a MIGRATE rate significantly lower than the XSTORE rate. However, there are times the MIGRATE value will spike for brief periods of time.
- Minidisk cache (MDC) statistics are given on the third line. The effectiveness of MDC can be judged by the combination of the READS rate and the HIT RATIO. If both are high, then a large number of physical I/Os are avoided due to the MDC feature. However, a high HIT RATIO with a low value for the READS rate is not good (it doesn't matter much if you have a 100% hit ratio, but are doing only 1 I/O per second).
- Line 4 describes more storage (memory) management. The PAGING rate is important. Higher values will often impact performance. The STEAL percentage is often misleading. This is basically the percentage of pages taken from guests that z/VM believes are non-dormant. Since some guests have periodic timers going off, they appear to be active to z/VM even when relatively idle. Pages taken from these guests are considered stolen. So there are scenarios where a system only has active guests, in which case all pages

taken would be considered stolen. Bearing this in mind, if a high STEAL value is observed, the paging rate needs to be checked. If the paging rate is low, then the STEAL value is not important.

- On lines 5 through 8 you also see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into 3 different classes (1 through 3) and a special class of zero. So the Column of Q_x values and E_x represent the virtual machines in the dispatch list and the eligible list. The most important value here to validate is that there are no virtual machines in the Eligible list: E1, E2, E3; this implies z/VM has stopped dispatching some virtual machines to avoid over committing resources. Do not worry about the values in parenthesis.

INDICATE QUEUES EXP

Another useful command to understand the state of the system is the **INDICATE QUEUES EXP**. 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 (a single virtual machine may have multiple virtual processors) what queue (dispatch list, eligible, limit list) they are in and what state they are. This is a snapshot in time. Again you want to make sure there are not any virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Q_x (x=1,2,3). Eligible list would be marked as E_x. The third column in the example also gives state of virtual processor. This can be helpful to get a feel for how the virtual processors might be constrained. Virtual processors that are actually running at the snapshot are marked with and RNN where NN is the processor number they are on. An R without a number means the virtual processor is ready to run but there is not an available processor. (**Note**: the virtual machine that issues the **INDICATE** command will always be one of the running machines). Other states are documented in the help for **IND Q EXP**. One doesn't have to be concerned about the other columns unless detailed analysis is required or if IBM support requests it. Also, always remember that is just a snapshot in time so often repeating this command over time can give a more accurate picture of your z/VM system.

13.1.2 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’s hard to find help for exactly the command you’re looking for. Some useful help commands are as follow

```
==> help             // for basic help
==> help cp menu     // for a menu of all CP commands
==> help cpquery     // for a menu of all CP QUERY command
==> help cpset       // for a menu of all CP SET commands
```

Determining who is logged on

To see who is logged on to the system use the **QUERY NAMES** command. For example

```
==> q n
LINUX06  - DSC , LINUX04  - DSC , LINUX03  - DSC , LINUX07  - DSC
LINUX01  - DSC , SLES9    - DSC , FTPSERVE - DSC , DTCVSW2  - DSC
DTCVSW1  - DSC , TCPIP    - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP     - DSC , OPERATOR - DSC , MAINT    -L0005
VSM     - TCPIP
```

Determining storage or memory

To see how much central and expanded storage (memory) use the **QUERY STORAGE** and **QUERY XSTOR** commands. For example:

```
==> q stor
STORAGE = 3G
==> q xstor
XSTORE= 1024M online= 1024M
XSTORE= 1024M userid= SYSTEM usage= 97% retained= 0M pending= 0M
XSTORE MDC min=0M, max=1024M, usage=96%
XSTORE= 1024M userid=  (none)  max. attach= 1024M
```

Determining processors or CPUs

To see how many processors (CPs, IFLs, CPUs) you have, use the **QUERY PROCESSORS** command. For example:

```
==> q proc
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
PROCESSOR 02 ALTERNATE
PROCESSOR 03 ALTERNATE
```

Determining software level

To determine what level of CP your system is at, use the **QUERY CPLEVEL** command. For example:

```
==> q cplevel
z/VM Version 5 Release 1.0, service level 0401 (64-bit)
Generated at 08/31/04 17:33:32 EST
IPL at 03/10/05 14:42:02 EST
```

Determining system cylinder allocation

The **QUERY ALLOC MAP** command shows you the system’s allocate of spool, paging and directory space. For example:

```
==> q alloc map
```

```
                EXTENT     EXTENT                         % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
520RES A700          1         20     20      1      1   5% DRCT ACTIVE
520SPL A703          0       3338 601020  40950  63360   6% SPOOL
520PAG A704          0       3338 601020   8167   9872   1% PAGE
MPA706 A706          0       3338 601020   7840  12448   1% PAGE
...
```

Determining DASD, OSA and virtual resources

The **QUERY DASD** and **QUERY DASD FREE** commands will show you what DASD is assigned to the system and what DASD is free to be assigned. Similarly the **QUERY OSA** and **QUERY OSA FREE** commands will report on the OSA resources. Finally, the **QUERY VIRTUAL ALL** command can be useful. Following is the short form of these commands without any output shown:

```
==> q da
==> q da free
==> q osa
==> q osa free
==> q v all
```

13.2 The z/VM Performance Toolkit

To use the z/VM Performance Toolkit, the product must be ordered. You should only configure the product if you have ordered it.

Much more detail can be found in the following books:

- *z/VM Performance Toolkit*, SC24-6136, on the Web starting at the z/VM 5.2 bookshelf:

  http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/zvmpdf/zvm52.html

  Search for **Toolkit** on that page.
- *The Program Directory for Performance Toolkit for VM*, GI11-2854-00, on the Web at

  http://www.vm.ibm.com/progdir/5vmptk20.pdf
- The IBM Redbook *Linux on IBM zSeries and S/390: Performance Toolkit for VM*, SG24-6059, on the Web at:

  http://www.redbooks.ibm.com/abstracts/sg246059.html

The section that follow describe how to set up and use the IBM Performance Toolkit *very briefly*:

- “Configuring the z/VM Performance Toolkit” on page 201
- “Using the z/VM Performance Toolkit” on page 205

13.2.1 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. You can query which priced products are enabled via the **QUERY PRODUCT** command:

```
==> q product
DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z
Ready; T=0.01/0.01 11:35:43
Product  State    Description
5VMDIR10 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FL 510
```

```
5VMPTK20 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
5684096K Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking Version 3 Release 2
 Modification 0
5767002P Disabled 00/00/00.00:00:00.$BASEDDR RACF for VM/ESA V2
```

To enable it, **logon to MAINT** and enter the following command:

```
==> service perftk enable
VMFSRV2760I SERVICE processing started
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

You should see a few screens of messages scroll by and finally the success messages shown above. This will enable the Performance Toolkit for the current z/VM session. To enable it across IPLs, modify the `SYSTEM CONFIG` file and uncomment the two lines changing `DISABLED` to **`ENABLED`** and reIPL your system:

```
==> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
==> link * cf1 cf1 mr
==> acc cf1 f
==> x system config f
====> /5vmptk10
PRODUCT PRODID 5VMPTK10 STATE ENABLED DESCRIPTION ‘00/00/00.00:00:00.$BASEDDR P
ERFORMANCE TOOLKIT FOR VM’
...
```

Check your changes for syntax errors and put the CP disk back online

```
==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> rel f (det
DASD 0CF1 DETACHED
==> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.
```

13.2.2 Configuring Web Browser support

Once the product is enabled, the TCPIP profile must be modified to add browser capabilities for the Performance Toolkit.

1. Logon to `TCPMAINT`. Edit the `<systemID> TCPIP D` file and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved

   ```
   ==> x <systemID> tcpip d
   ====> /port
   ```

2. Add the following line under the PORT entries:

   ```
   ...
   PORT
     20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
     21   TCP FTPSERVE            ; FTP Server
     23   TCP INTCLIEN            ; TELNET Server
   ; 25   TCP SMTP                ; SMTP Server
   ; 53   TCP NAMESRV             ; Domain Name Server
   ; 53   UDP NAMESRV             ; Domain Name Server
   ; 67   UDP BOOTPD              ; BootP Server
   ```

```
; 67   UDP DHCPD            ; DHCP Server
; 69   UDP TFTPD            ; TFTPD (Trivial FTP) Server
  81   TCP PERFSVM          ; Performance Toolkit
; 111  TCP PORTMAP          ; Portmap Server
...
```

3. Save your changes. The `TCPIP` user ID needs to be recycled in order for our changes to take effect. You can **`FORCE`** and **`XAUTOLOG TCPIP`** from a console. Alternatively, **if you are in postion** to reIPL the system, you can issue the command **`shutdown reipl iplparms cons=sysc`**.

When the system comes back, logon to `TCPMAINT` and check if everything was successful by issuing the **`NETSTAT CLIENTS`** command. You want to see that the service `PERFSVM` is a client (listening). This should be shown after a few screens of output:

```
==> netstat clients
...
Client: PERFSVM                Authorization: {none}
Notes Handled: none
Last Touched:   0:00:25
Vmcf error count: 0
```

The output of the command is lengthy, you want to make sure you have an entry for `PERFSVM`.

13.2.3 Configuring PERFSVM

The `PERFSVM` user ID is the Performance Toolkit service machine.

1. **Logon to `PERFSVM`**. If you successfully enabled the product, you should be put in a Performance Toolkit session and see the following text at the top of the screen:

```
FCX001                 Performance Toolkit for VM
FCXBAS500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

2. Press **PF12 twice** to get to a CMS prompt.
3. Copy the `PROFILE XEDIT` from the `MAINT 191` disk so XEDIT sessions will have a common interface among user IDs.
   a. Use the **`VMLINK`** command to both link the disk read-only and access it as the highest available file mode. The default read password is **read,** however, if you changed your passwords as described in section 4.9.1, “Changing passwords in USER DIRECT” on page 60, then it will be `lnx4vm`.

```
==> vmlink maint 191
ENTER READ PASSWORD:
lnx4vm
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

   b. Copy the `PROFILE XEDIT` to the A disk:

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

4. Copy the default configuration files, which are on `PERFSVM`'s D disk, to your A disk:

```
==> copy * * d = = a
```

5. The main configuration file is `FCONX $PROFILE`. Edit that file and search for the string VMCF. This should take you to line 173 where the next 4 lines are comments starting with an *. Perform the following changes:

– Uncomment the second and fourth line by changing `*C` to **`FC`**
– Change `IDTEST PASSFILE` to `IDTEST` **`CP`** on the fourth line

- Add the text **`FC MONCOLL LINUXUSR ON`** after the fourth line

The modified lines should be as follows:

```
==> x fconx $profile a
====> /vmcf
*    Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
*    Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 81 IDTEST CP
FC MONCOLL LINUXUSR ON
*    Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
```

6. Save your changes. The line you added tells the Performance Toolkit to collect Linux performance data.
7. Create a remote data retrieval authorization file – replace `<systemID>` with your system name:

```
==> x fconrmt authoriz
====> a 2
<systemID>  PERFSVM  S&FSERV
<systemID>  MAINT    DATA CMD EXCPMSG
```

8. Create a system identification file – replace `<systemID>` with your z/VM system name:

```
==> x fconrmt systems
====> a
<systemID>   PERFSVM  ESA   N   FCXRES00
```

9. Create a Linux system definition file. Add the TCP/IP addresses of your Linux system(s). The following example shows adding seven Linux virtual servers:

   Use port 8803 for Linux performance data:

```
==> x fconx linuxusr
LINUX01 129.40.178.121:8803
LINUX02 129.40.178.122:8803
LINUX03 129.40.178.123:8803
LINUX04 129.40.178.124:8803
LINUX05 129.40.178.125:8803
LINUX06 129.40.178.126:8803
LINUX07 129.40.178.127:8803
```

10. Edit the `PROFILE EXEC` file and uncomment the five `MONITOR SAMPLE` and the two `MONITOR EVENT` statements:

    Before:

```
...
/*** Once you have PERFKIT enabled and running uncomment the      ***/
/*** following comments                                           ***/
/* 'CP MONITOR SAMPLE ENABLE PROCESSOR'    */
/* 'CP MONITOR SAMPLE ENABLE STORAGE'      */
/* 'CP MONITOR SAMPLE ENABLE USER ALL'     */
/* 'CP MONITOR SAMPLE ENABLE I/O ALL'      */
/* 'CP MONITOR SAMPLE ENABLE APPLDATA ALL' */
/* 'CP MONITOR EVENT  ENABLE STORAGE'      */
/* 'CP MONITOR EVENT  ENABLE I/O ALL'      */

'PERFKIT'                    /* Invoke the PERFKIT module   @FC012BD*/

Exit
```

After:

```
...
/*** Once you have PERFKIT enabled and running uncomment the      ***/
/*** following comments                                           ***/
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'CP MONITOR SAMPLE ENABLE STORAGE'
'CP MONITOR SAMPLE ENABLE USER ALL'
'CP MONITOR SAMPLE ENABLE I/O ALL'
'CP MONITOR SAMPLE ENABLE APPLDATA ALL'
'CP MONITOR EVENT  ENABLE STORAGE'
'CP MONITOR EVENT  ENABLE I/O ALL'

'PERFKIT'                    /* Invoke the PERFKIT module   @FC012BD*/

Exit
```

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

You should now be ready to run the Performance Toolkit.

13.2.4 Starting the z/VM Performance Toolkit

To start the Performance Toolkit enter the following command:

```
===> perfkit
 FCX001                 Performance Toolkit for VM                Autoscroll 12
  FCXBAS500I Performance Toolkit for VM FL520 BASE
  FCXAPP530I Connected to *IDENT for resource FCXRES00
  FCXAPF530I Connected to *IDENT for resource FCXSYSTM
  FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
  FCXTCR571I Connected to TCP/IP server TCPIP on path 0004
  FCXAPP527I User PERFSVM connected on path 0007
  FCXAPC535I Connected to resource FCXRES00 on path 0006, for S&F-Coll
  FCXTCP575I WebServer host IP address is 129.40.178.120:00081
  FCXTCR575I TCPRequest host IP address is 129.40.178.120:41780
  FCXTCR590I TCPRequest interface activated
  HCPMOF6229E Monitor event collection is already active.
  HCPMOG6229E Monitor sample collection is already active.
  FCXTCP590I WebServer interface activated
```

Disconnect from PERFSVM now.

```
Command ===> disc
```

The Performance Toolkit should now be configured and running.

13.2.5 Using the z/VM Performance Toolkit

The Performance Toolkit can be used via a Web browser or 3270 interface.

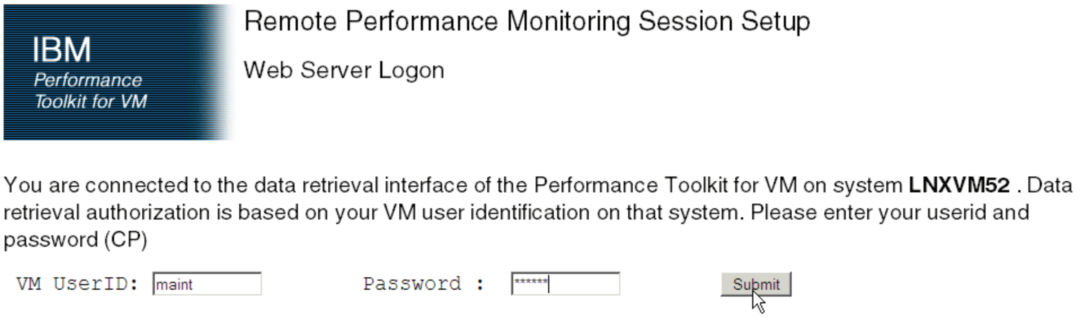
Using a Web browser interface

To use the Web-enabled Performance Toolkit, perform the following steps:

1. Point a browser to your z/VM system at port 81. For example:

   ```
   http://129.40.178.120:81
   ```

2. You should see your system on the *Web Session Setup* screen. Click it and you will be presented with the Web Server Logon screen.



Remote Performance Monitoring Session Setup

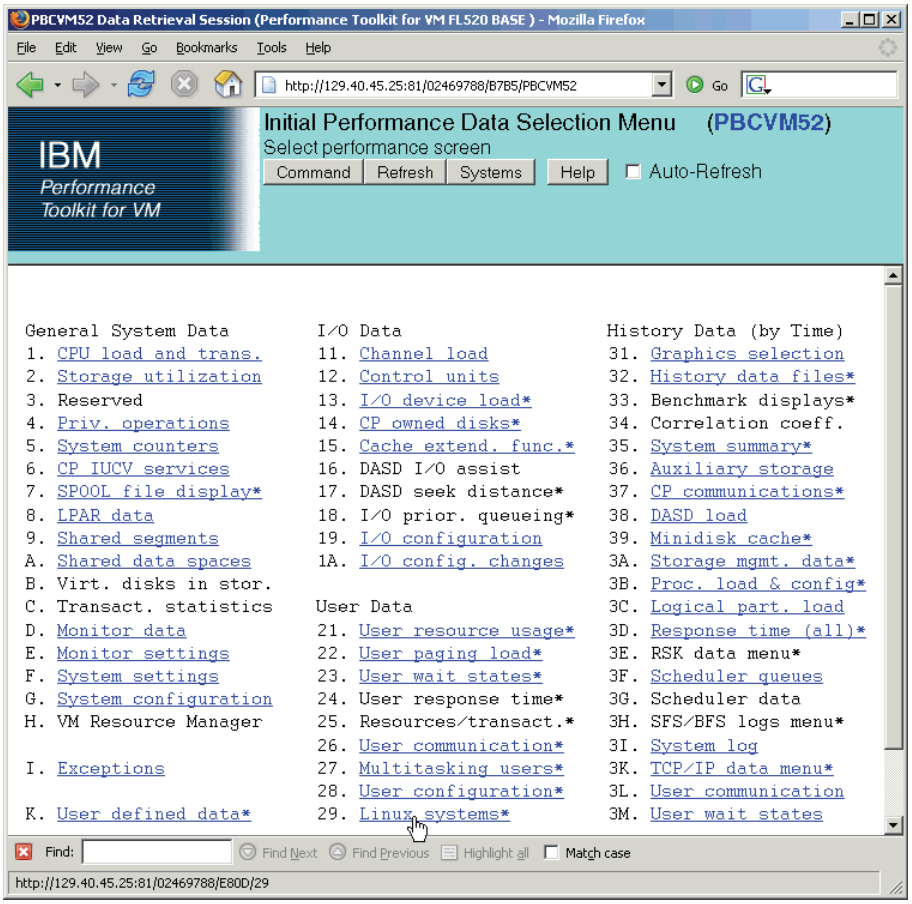
Web Server Logon

You are connected to the data retrieval interface of the Performance Toolkit for VM on system **LNXVM52** . Data retrieval authorization is based on your VM user identification on that system. Please enter your userid and password (CP)

VM UserID: maint   Password : ******   Submit

Figure 13-1 Performance Toolkit logon screen

3. Enter any valid user ID and password. `MAINT` can be used.
4. You should see the *Central Monitoring System Load Overview* with your system name on the left side.
5. Click your system name and you should see the *Initial Performance Data Selection Menu* screen as shown in Figure 13-2 on page 206.



PBCVM52 Data Retrieval Session (Performance Toolkit for VM FL520 BASE) - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.45.25:81/02469788/B7B5/PBCVM52
Go
IBM
Performance
Toolkit for VM
Command
Refresh
Systems
Help
□ Auto-Refresh
Find:
Find Next
Find Previous
Highlight all
□ Match case

Initial Performance Data Selection Menu (**PBCVM52**)

Select performance screen

```
General System Data          I/O Data                   History Data (by Time)
1. CPU load and trans.       11. Channel load           31. Graphics selection
2. Storage utilization       12. Control units          32. History data files*
3. Reserved                  13. I/O device load*       33. Benchmark displays*
4. Priv. operations          14. CP owned disks*        34. Correlation coeff.
5. System counters           15. Cache extend. func.*   35. System summary*
6. CP IUCV services          16. DASD I/O assist        36. Auxiliary storage
7. SPOOL file display*       17. DASD seek distance*    37. CP communications*
8. LPAR data                 18. I/O prior. queueing*   38. DASD load
9. Shared segments           19. I/O configuration      39. Minidisk cache*
A. Shared data spaces        1A. I/O config. changes    3A. Storage mgmt. data*
B. Virt. disks in stor.                                 3B. Proc. load & config*
C. Transact. statistics      User Data                  3C. Logical part. load
D. Monitor data              21. User resource usage*   3D. Response time (all)*
E. Monitor settings          22. User paging load*      3E. RSK data menu*
F. System settings           23. User wait states*      3F. Scheduler queues
G. System configuration      24. User response time*    3G. Scheduler data
H. VM Resource Manager       25. Resources/transact.*   3H. SFS/BFS logs menu*
                             26. User communication*    3I. System log
I. Exceptions                27. Multitasking users*    3K. TCP/IP data menu*
                             28. User configuration*    3L. User communication
K. User defined data*        29. Linux systems*         3M. User wait states
```

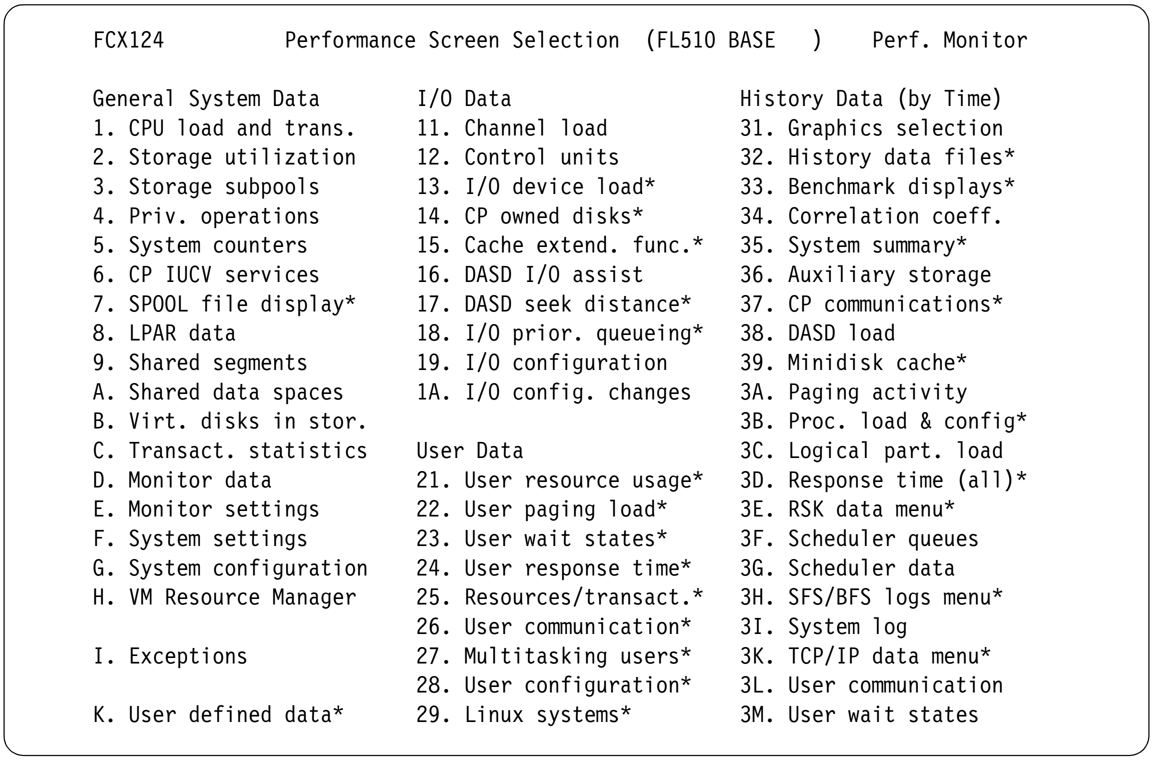
http://129.40.45.25:81/02469788/E80D/29

Figure 13-2 Browser interface to the Performance Toolkit

Using a 3270 interface

Logon to PERFSVM. Run the **PROFILE EXEC** and you should be put into the Performance Toolkit for z/VM environment. The subcommand **monitor** should present the panel shown below.

```
==> profile
FCXBAS500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
...
FCX001                 Performance Toolkit for VM
  FCXBAS500I Performance Toolkit for VM FL510 BASE
  HCPMOF6229E Monitor event collection is already active.
  HCPMOG6229E Monitor sample collection is already active.

Command ==> monitor
```



```
FCX124         Performance Screen Selection  (FL510 BASE   )    Perf. Monitor

General System Data       I/O Data                  History Data (by Time)
1. CPU load and trans.    11. Channel load          31. Graphics selection
2. Storage utilization    12. Control units         32. History data files*
3. Storage subpools       13. I/O device load*      33. Benchmark displays*
4. Priv. operations       14. CP owned disks*       34. Correlation coeff.
5. System counters        15. Cache extend. func.*  35. System summary*
6. CP IUCV services       16. DASD I/O assist       36. Auxiliary storage
7. SPOOL file display*    17. DASD seek distance*   37. CP communications*
8. LPAR data              18. I/O prior. queueing*  38. DASD load
9. Shared segments        19. I/O configuration     39. Minidisk cache*
A. Shared data spaces     1A. I/O config. changes   3A. Paging activity
B. Virt. disks in stor.                             3B. Proc. load & config*
C. Transact. statistics   User Data                 3C. Logical part. load
D. Monitor data           21. User resource usage*  3D. Response time (all)*
E. Monitor settings       22. User paging load*     3E. RSK data menu*
F. System settings        23. User wait states*     3F. Scheduler queues
G. System configuration   24. User response time*   3G. Scheduler data
H. VM Resource Manager    25. Resources/transact.*  3H. SFS/BFS logs menu*
                          26. User communication*   3I. System log
I. Exceptions             27. Multitasking users*   3K. TCP/IP data menu*
                          28. User configuration*   3L. User communication
K. User defined data*     29. Linux systems*        3M. User wait states
```

Drilling down into report screens

You should now be able to use the active report screens. To drill down into these screens, move the cursor to any of the titles that are active (active titles display the number or letter in white, inactive titles are in green). Some of the more useful report screens to drill down into are:

```
21. User resource usage
22. User paging load
23. User wait states
28. User configuration
29. Linux systems
```

13.3 Monitoring Linux

To monitor Linux performance data, a data gatherer process must be running. There are different ways of gathering this data. Data can be gathered in the kernel and as a user application. SuSE SLES9, SLES10 and Red Hat RHEL 4 and RHEL 5 have been enabled for the kernel to gather performance data. There is a package called the *Linux RMF PM Data Gatherer* (also called *rmfpms*) that runs as a user application. Both of these data gatherers work in conjunction with the IBM z/VM Performance Toolkit.

13.3.1 Monitoring Linux with rmfpms

As a user application, the Linux RMF PM Data Gatherer (rmfpms) can be used. Currently it is not part of an IBM product and is intended for evaluation purposes only. A description of rmfpms is as follows:

> "rmfpms is a modular data gatherer for Linux®. The gathered data can be analyzed using the RMF PM client application. The performance data is accessible through XML over HTTP so you can easily exploit it in your own applications."

The following Web site is a starting point:

```
http://www-03.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.html
```

To download the data gatherer, scroll down and look for the following text and links:

```
* 31 bit data gatherer (kernel24 - 630 KB, kernel26 - 1040 KB).
* 64 bit data gatherer (kernel24 - 650 KB, kernel26 - 666 KB).
```

You can download the appropriate gatherer via a browser, or, if you have access to the Internet you can use an FTP client. You will want one of two files depending on whether you have a 31-bit or 64-bit kernel:

```
rmfpms_s390_kernel26.tgz - for 31-bit distributions
rmfpms_s390x_kernel26.tgz - for 64-bit distributions
```

Following is an example of downloading the tar file for 31-bit distributions directly from the Internet. **Get an SSH session on the controller** or any other virtual server. Change to the home directory via the cd command with no argument. Download the appropriate tar file with the **wget** command. For a 64-bit SLES 9 and for SLES 10 use:

```
# cd
# wget ftp://ftp.software.ibm.com/eserver/zseries/zos/rmf/rmfpms_s390x_kernel26.tgz
```

For RHEL 4 and RHEL 5 use:

```
# wget ftp://ftp.software.ibm.com/eserver/zseries/zos/rmf/rmfpms_s390x_kernel26_rhel4.tgz
```

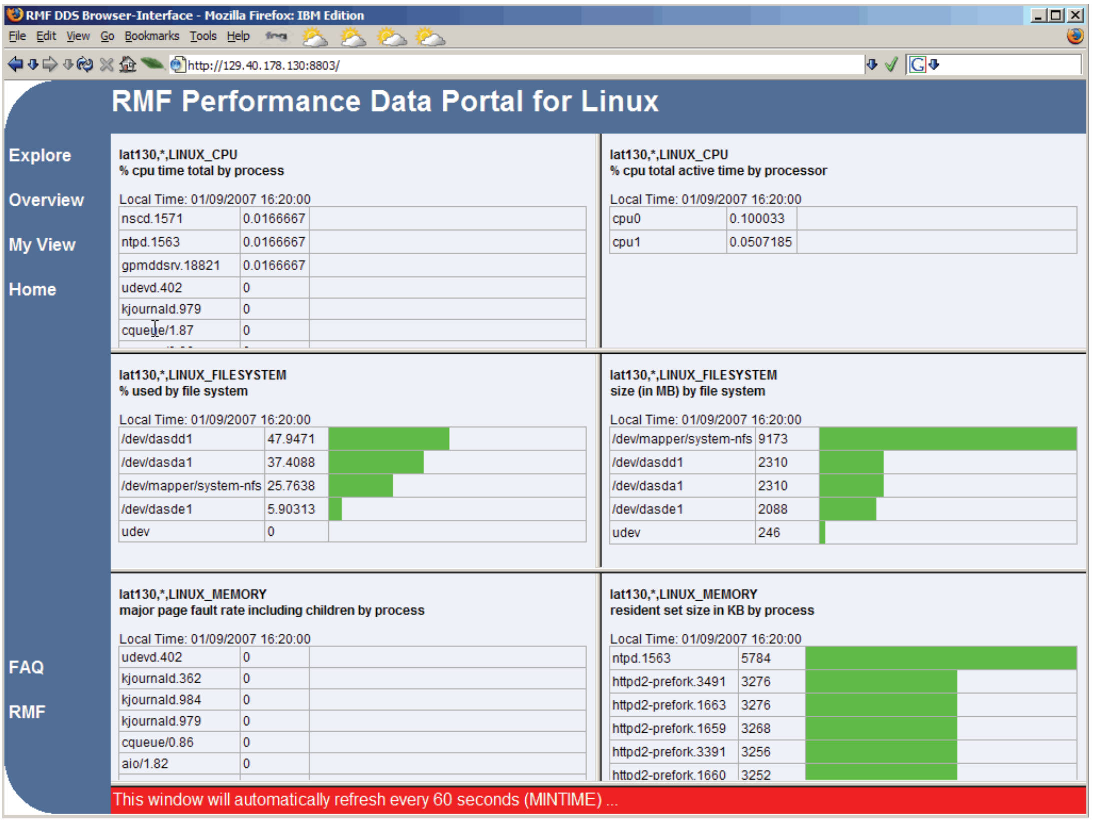
Untar the file with the **tar** command and change to the `rmfpms/` directory:

```
# tar xzf rmfpms_s390x_kernel26*.tgz
# cd rmfpms/
# ls -aF
./   .rmfpms_config            README           bin/  enable_autostart*
../  .rmfpms_config_autostart  autostart_rmfpms  doc/
```

You should now be able to start **rmfpms** in the `bin/` directory with the following command:

```
# bin/rmfpms start
Creating /opt/IBM/rmfpms/.rmfpms ...
Starting performance gatherer backends ...
 DDSRV: RMF-DDS-Server/Linux-Beta (Aug  9 2004) started.
 DDSRV: Functionality Level=2.008
 DDSRV: Reading exceptions from gpmexsys.ini and gpmexusr.ini.
DDSRV: Server will now run as a daemon process.
done!
```

Once it is running, you can view the performance data from a browser pointing to the Linux image and port 8803 as shown in Figure 13-3.


RMF DDS Browser-Interface - Mozilla Firefox: IBM Edition
File Edit View Go Bookmarks Tools Help
http://129.40.178.130:8803/
Explore
Overview
My View
Home
FAQ
RMF

RMF Performance Data Portal for Linux

lat130,*,LINUX_CPU
% cpu time total by process

Local Time: 01/09/2007 16:20:00

nscd.1571	0.0166667
ntpd.1563	0.0166667
gpmddsrv.18821	0.0166667
udevd.402	0
kjournald.979	0
cqueue/1.87	0

lat130,*,LINUX_CPU
% cpu total active time by processor

Local Time: 01/09/2007 16:20:00

cpu0	0.100033
cpu1	0.0507185

lat130,*,LINUX_FILESYSTEM
% used by file system

Local Time: 01/09/2007 16:20:00

/dev/dasdd1	47.9471
/dev/dasda1	37.4088
/dev/mapper/system-nfs	25.7638
/dev/dasde1	5.90313
udev	0

lat130,*,LINUX_FILESYSTEM
size (in MB) by file system

Local Time: 01/09/2007 16:20:00

/dev/mapper/system-nfs	9173
/dev/dasdd1	2310
/dev/dasda1	2310
/dev/dasde1	2088
udev	246

lat130,*,LINUX_MEMORY
major page fault rate including children by process

Local Time: 01/09/2007 16:20:00

udevd.402	0
kjournald.362	0
kjournald.984	0
kjournald.979	0
cqueue/0.86	0
aio/1.82	0

lat130,*,LINUX_MEMORY
resident set size in KB by process

Local Time: 01/09/2007 16:20:00

ntpd.1563	5784
httpd2-prefork.3491	3276
httpd2-prefork.1663	3276
httpd2-prefork.1659	3268
httpd2-prefork.3391	3256
httpd2-prefork.1660	3252

This window will automatically refresh every 60 seconds (MINTIME) ...

Figure 13-3   Browser view of rmfpms interface

You can also register Linux images with the Performance Toolkit - see section 13.4, “Registering Linux images with the Performance Toolkit” on page 210.

13.3.2 Monitoring Linux performance data from the kernel

To monitor Linux performance data directly from the kernel, the following must be true:

1. The APPLMON option must be set in the user directory.
2. Applmon data monitoring must be built into the kernel.

The first requirement should be true as the OPTION APPLMON was set for the controller and for Linux user IDs in earlier sections.

For the second requirement, SLES 9, SLES 10 and RHEL 5 now have this function built in. Details of this function are described in Chapter 15, *Linux monitor stream support for z/VM* in the manual *Device Drivers, Features, and Commands* for the October 2005 stream, on the Web at:

http://www-128.ibm.com/developerworks/linux/linux390/october2005_documentation.html

A quick description of how to use this built-in monitoring function follows.

There are three modules that are built into the kernel but are not loaded by default. They are named appldata_mem, appldata_os and appldata_net_sum. You can verify that they are not loaded with the **lsmod** and **grep** commands:

```
# lsmod | grep appldata
```

There is no output so no modules with the string appldata are loaded. Load those modules now via the **modprobe** command and verify they have been loaded:

```
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

Now if you repeat the **lsmod** command, you should see the following:

```
# lsmod | grep appldata
appldata_net_sum      20064  0
appldata_os           21512  0
appldata_mem          20112  0
```

The directory in the virtual /proc/ file system where the monitoring variables exist is /proc/sys/appldata/. In this directory there are five files as follow:

timer	Controls whether any data gathering is in effect.
interval	Sets the interval, in milliseconds, that samples will be taken.
mem	Controls the memory data gathering module
os	Controls the CPU data gathering module
net_sum	Controls the net data gathering module

To turn on the built in kernel monitoring, use the **echo** command to send a non-zero value into four of the five monitoring variables in the /proc/ virtual file system:

```
# echo 1 > /proc/sys/appldata/timer
# echo 1 > /proc/sys/appldata/mem
# echo 1 > /proc/sys/appldata/os
# echo 1 > /proc/sys/appldata/net_sum
```

Built-in kernel monitoring should now be turned on.You 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.

13.4 Registering Linux images with the Performance Toolkit

To register Linux images that have performance data gathering enabled, **logon to PERFSVM** and create a file FCONX LINUXUSR A and add Linux user ID/IP address:port pairs. The following example shows adding the SLES9X and LINUX01 user IDs to this file:

```
==> x fconx linuxusr a
====> a 3
*Linux-ID    IP address for DDS Interface:Port
SLES9X       <129.40.178.130>
LINUX01      <129.40.178.121>
```

Restart the performance toolkit via the **profile** command. After the system has had some time to collect data, you should be able to use the Performance Toolkit to monitor Linux systems that have both monitoring data being captured and an entry in the FCONX LINUXUSR file. To view that data, drill down into menu 29, Linux systems.

13.5 Monitoring z/VM and Linux via a Web application

A basic Web application is written to demonstrate one way to view data of both z/VM and the Linux virtual servers. The system is named DAZL, an acronym for **D**ata **A**bout **z**/VM and **L**inux (yes, a name shamelessly force fit to come up with a cool acronym :)). To supply the DAZL information, a Web Server is run on the controller that makes available some HTML pages in the DocumentRoot directory and some bash shell scripts in the `cgi-bin/` directory. If you have been progressing sequentially through the book, the Web server should be configured by now. If not, see chapter 7, especially the section “Configuring Apache for DAZL” in section 7.7.

A philosophy of DAZL is to keep common information grouped into pages about one screen in size, utilizing HTML tables to maximize the real estate of a browser screen. This application is not a heavily tested, nor is it a professional monitoring system. Rather it is a set of simple Web pages and scripts that can be a start of a home-grown monitoring tool, if you choose to investigate it.

This section shows some screen captures of the Web application, then the application’s relatively simple design is described in some detail in section 13.5.3, “DAZL design” on page 220. All source code is included in the controller RPM described in.

Important: DAZL is not a strategic direction for monitoring z/VM and Linux. A more strategic solution from IBM’s point of view is IBM Director and z/VM Center. See section 1.4, “IBM Director and z/VM Center Extension” on page 5.

13.5.1 Overall design

Following is a block diagram of the overall design of DAZL from an execution point of view:

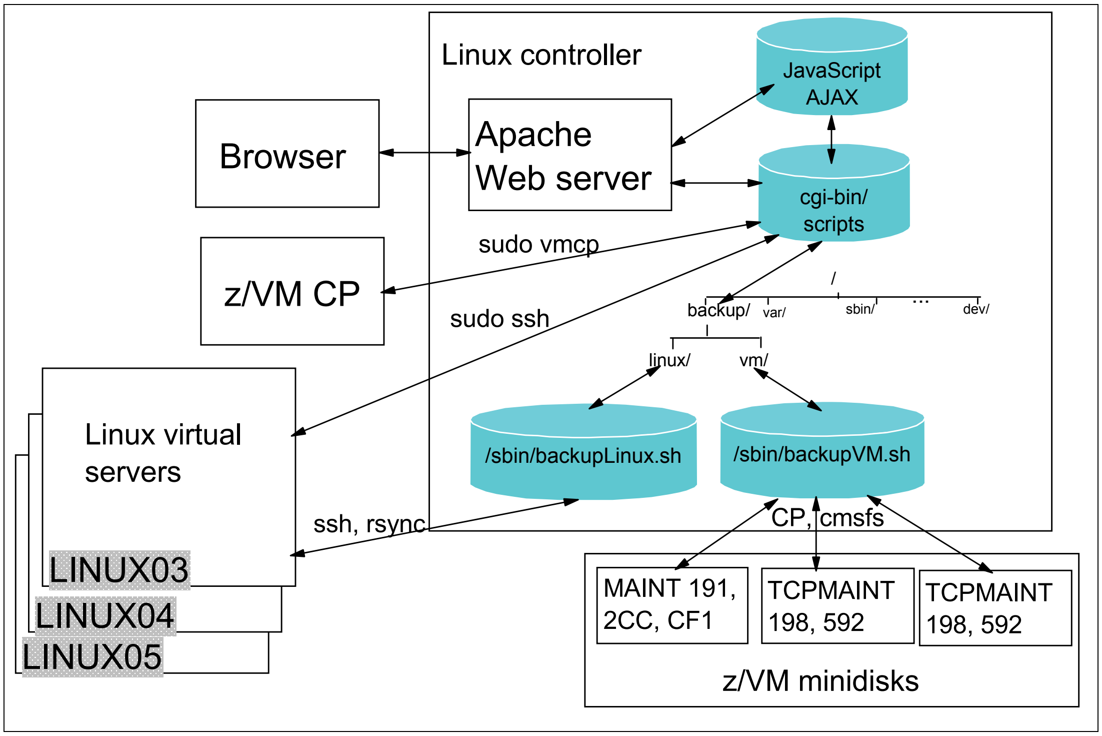


Figure 13-4 DAZL overall design

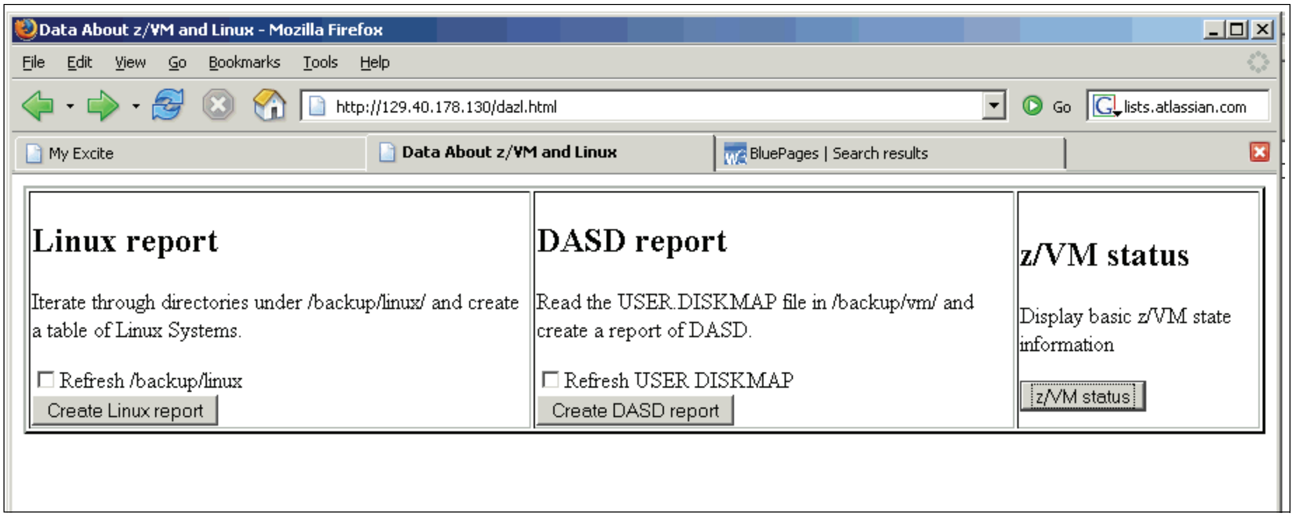
13.5.2 DAZL screen captures

The main Web page is shipped with the file name `index.html`. If you followed all the steps in section It allows you to generate basic pages or reports about:

Linux systems — Each Linux virtual server that was cloned
z/VM DASD — Each volume in the `USER DISKMAP` report on the `MAINT 191` disk
z/VM status — Output of simple z/VM query commands

DAZL home page

Following is a screen capture of the DAZL home page:


Data About z/VM and Linux - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
http://129.40.178.130/dazl.html
Go
lists.atlassian.com
My Excite
Data About z/VM and Linux
BluePages | Search results

Linux report

Iterate through directories under /backup/linux/ and create a table of Linux Systems.

□ Refresh /backup/linux

Create Linux report

DASD report

Read the USER.DISKMAP file in /backup/vm/ and create a report of DASD.

□ Refresh USER DISKMAP

Create DASD report

z/VM status

Display basic z/VM state information

z/VM status

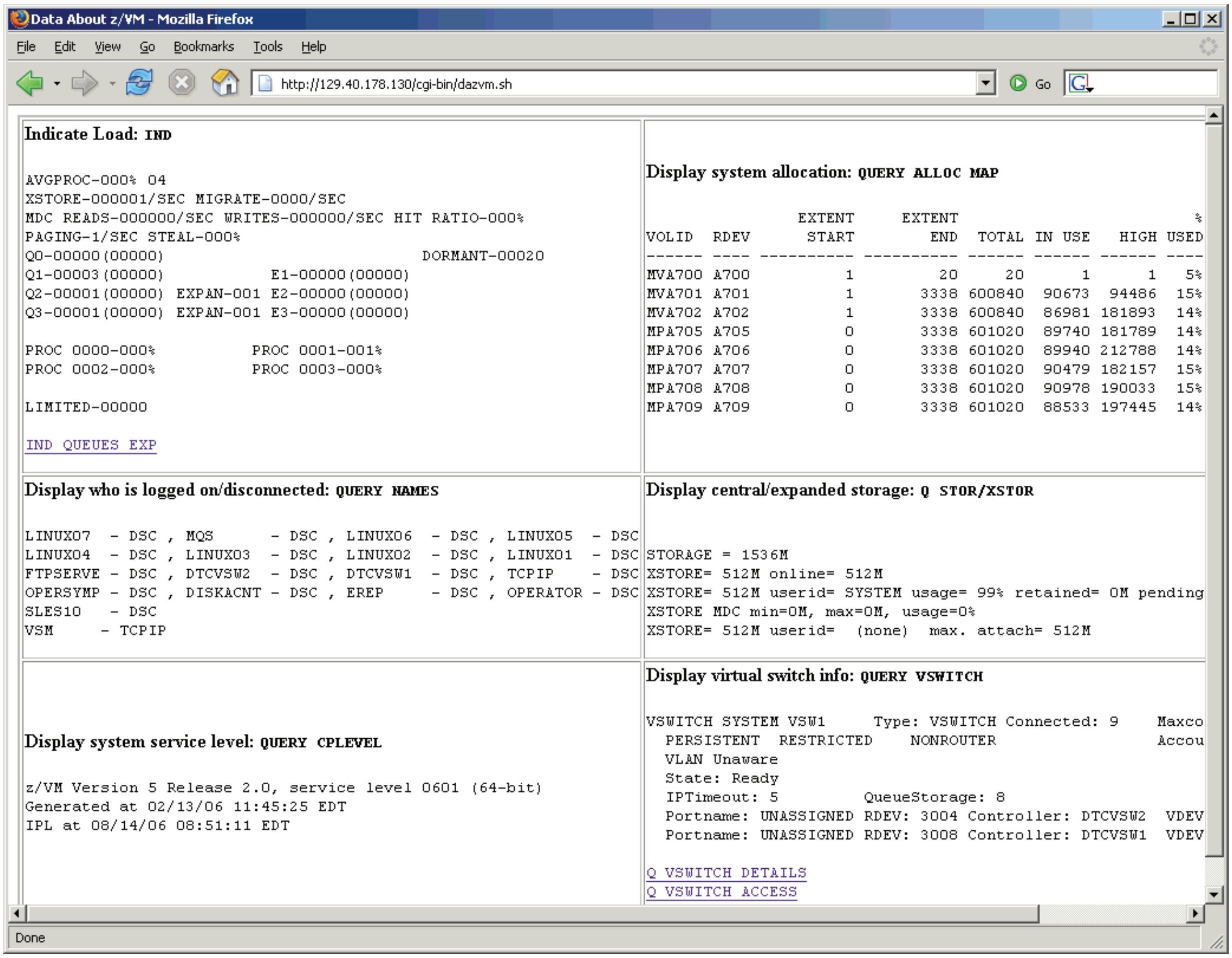
Figure 13-5 DAZL main page

z/VM status page

If you click the **z/VM status** button on the right, you will get some basic information about your z/VM system from the following commands:

- `IND`
- `Q ALLOC MAP`
- `Q NAMES`
- `Q STOR, Q XSTOR`
- `Q CPLEVEL`
- `Q VSWITCH`

These values are obtained via the Linux **`vmcp`** interface to z/VM CP commands. Following is a screen capture of the z/VM status page:


Data About z/VM - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.130/cgi-bin/dazvm.sh
Go
Done

Indicate Load: **IND**

```
AVGPROC-000% 04
XSTORE-000001/SEC MIGRATE-0000/SEC
MDC READS-000000/SEC WRITES-000000/SEC HIT RATIO-000%
PAGING-1/SEC STEAL-000%
Q0-00000(00000)                           DORMANT-00020
Q1-00003(00000)           E1-00000(00000)
Q2-00001(00000)  EXPAN-001 E2-00000(00000)
Q3-00001(00000)  EXPAN-001 E3-00000(00000)

PROC 0000-000%          PROC 0001-001%
PROC 0002-000%          PROC 0003-000%

LIMITED-00000

IND QUEUES EXP
```

Display system allocation: **QUERY ALLOC MAP**

```
                  EXTENT     EXTENT                              %
VOLID  RDEV        START        END  TOTAL  IN USE   HIGH USED
------ ---- ---------- ---------- ------ ------ ------ ----
MVA700 A700          1         20     20      1      1   5%
MVA701 A701          1       3338 600840  90673  94486  15%
MVA702 A702          1       3338 600840  86981 181893  14%
MPA705 A705          0       3338 601020  89740 181789  14%
MPA706 A706          0       3338 601020  89940 212788  14%
MPA707 A707          0       3338 601020  90479 182157  15%
MPA708 A708          0       3338 601020  90978 190033  15%
MPA709 A709          0       3338 601020  88533 197445  14%
```

Display who is logged on/disconnected: **QUERY NAMES**

```
LINUX07  - DSC , MQS      - DSC , LINUX06  - DSC , LINUX05  - DSC
LINUX04  - DSC , LINUX03  - DSC , LINUX02  - DSC , LINUX01  - DSC
FTPSERVE - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , TCPIP    - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
SLES10   - DSC
VSM      - TCPIP
```

Display central/expanded storage: **Q STOR/XSTOR**

```
STORAGE = 1536M
XSTORE= 512M online= 512M
XSTORE= 512M userid= SYSTEM usage= 99% retained= 0M pending
XSTORE MDC min=0M, max=0M, usage=0%
XSTORE= 512M userid=  (none)  max. attach= 512M
```

Display system service level: **QUERY CPLEVEL**

```
z/VM Version 5 Release 2.0, service level 0601 (64-bit)
Generated at 02/13/06 11:45:25 EDT
IPL at 08/14/06 08:51:11 EDT
```

Display virtual switch info: **QUERY VSWITCH**

```
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 9    Maxco
  PERSISTENT  RESTRICTED    NONROUTER                 Accou
  VLAN Unaware
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW2  VDEV
  Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW1  VDEV

Q VSWITCH DETAILS
Q VSWITCH ACCESS
```

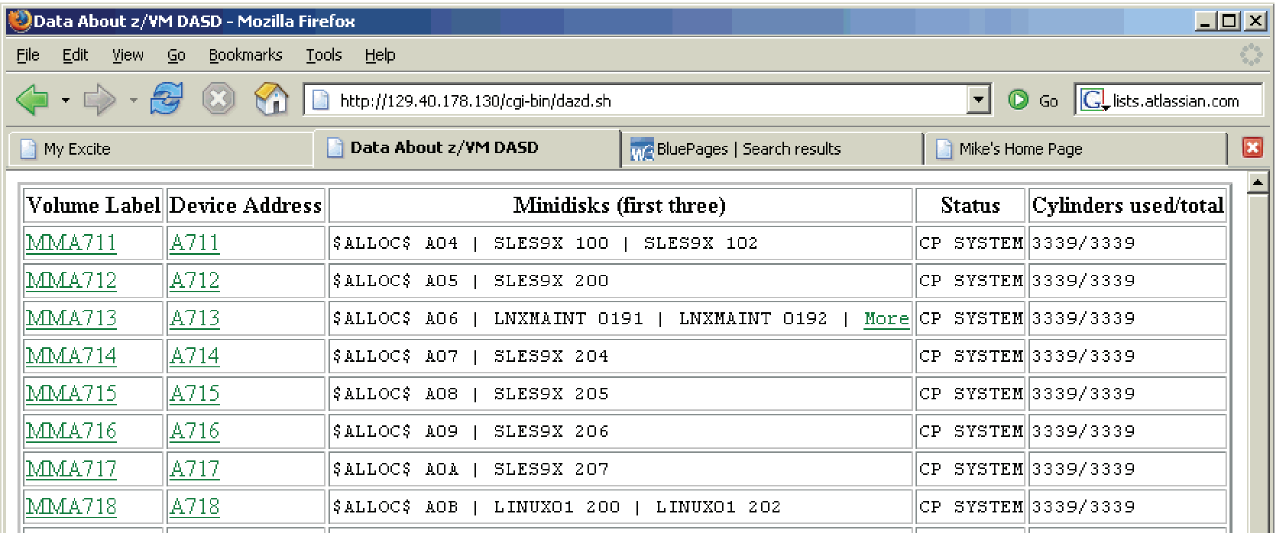
Figure 13-6 z/VM status page

From this page, you can drill down to get more detailed information. From the **Indicate Load** table cell, you can drill down to view the output of **`INDICATE QUEUES EXP`**. From the **QUERY VSWITCH** table cell, you can drill down to view the output of the **`QUERY VSWITCH DETAILS`** and **`QUERY VSWITCH ACCESS`** commands.

DASD report

If you click on the **DASD report** button in the center of the main page, you will get a report on the z/VM system volumes that comprise your system. This report is another way of displaying the contents of the `USER DISKMAP` file. The accuracy of the report generated relies on the `USER DISKMAP` file being up to date under `/backup/vm/USER.DISKMAP` on the controller. There is a checkbox that allows update this file on Linux from the `MAINT 191` disk. Clicking that checkbox will copy the `USER DISKMAP` file from z/VM, however, it still relies on the file being up to date. If you have modified the `USER DIRECT` file and run **`DIRECTXA`** but not run **`DISKMAP`**, then there is a good chance the data will be out of date (a good convention to adopt is to always run **`DISKMAP`** before running **`DIRECTXA`**).

Following is a screen capture of the DASD report


Data About z/VM DASD - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
http://129.40.178.130/cgi-bin/dazd.sh
Go
lists.atlassian.com
My Excite
Data About z/VM DASD
BluePages | Search results
Mike's Home Page

Volume Label	Device Address	Minidisks (first three)	Status	Cylinders used/total
MMA711	A711	$ALLOC$ A04 | SLES9X 100 | SLES9X 102	CP SYSTEM	3339/3339
MMA712	A712	$ALLOC$ A05 | SLES9X 200	CP SYSTEM	3339/3339
MMA713	A713	$ALLOC$ A06 | LNXMAINT 0191 | LNXMAINT 0192 | More	CP SYSTEM	3339/3339
MMA714	A714	$ALLOC$ A07 | SLES9X 204	CP SYSTEM	3339/3339
MMA715	A715	$ALLOC$ A08 | SLES9X 205	CP SYSTEM	3339/3339
MMA716	A716	$ALLOC$ A09 | SLES9X 206	CP SYSTEM	3339/3339
MMA717	A717	$ALLOC$ A0A | SLES9X 207	CP SYSTEM	3339/3339
MMA718	A718	$ALLOC$ A0B | LINUX01 200 | LINUX01 202	CP SYSTEM	3339/3339

Figure 13-7 DASD report

For each DASD in the report, you can drill down into the **Volume Label** column to get the text of that volume's report in the `USER DISKMAP` file, and into the **Device Address** column to get the output of the CP command **`QUERY DASD DETAILS <vaddr>`** for that volume.

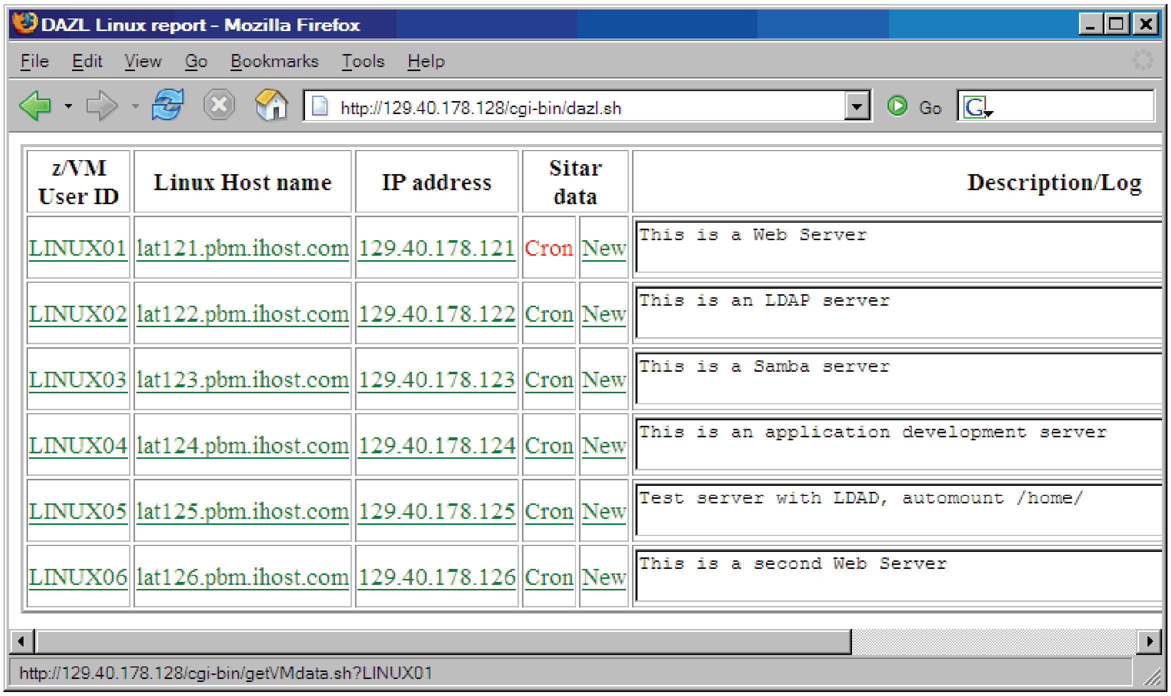
Linux report

If you click on **Linux report** in the left side of the main page, you will get a report on the Linux clones. It is generated starting from data under the `/backup/linux/` directory. When the **`clone.sh`** script clones a new Linux image, it also creates a directory of the form `<userID>-on-<TCP.IP.address>/` under `/backup/linux/`. For example, if you clone a virtual server to `LINUX01` with an IP address of 129.40.178.121, the directory created will be `/backup/linux/LINUX01-on-129.40.178.121/`. A row is created in the report's table for each directory, assuming they are associated with Linux systems. If this directory structure does not reflect the existing Linux images, the report will not be correct.

> **Important:** If you have not used the **`clone.sh`** script to clone and still want to try out DAZL, this is possible. You will just have to create a directory structure of Linux servers under `/backup/linux/` described above.

For each directory found, the user ID and TCP/IP address are extracted from the name. The existence of these directories drive the main loop of the report generation. Each TCP/IP address that is extracted from the directory structure is **`ping`**ed. If the ping is successful, the table cell entries for that virtual server are displayed in green and made to be drill down links. If the ping is not successful, the table cell entries are displayed as plain red text. The **Sitar Cron** entries are drawn in green if the `sitar.html` file is found in each virtual server's `/etc/` directory and in red if it is not.

Following is a screen capture of the Linux report:


DAZL Linux report - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.128/cgi-bin/dazl.sh
Go

z/VM User ID	Linux Host name	IP address	Sitar data		Description/Log
LINUX01	lat121.pbm.ihost.com	129.40.178.121	Cron	New	This is a Web Server
LINUX02	lat122.pbm.ihost.com	129.40.178.122	Cron	New	This is an LDAP server
LINUX03	lat123.pbm.ihost.com	129.40.178.123	Cron	New	This is a Samba server
LINUX04	lat124.pbm.ihost.com	129.40.178.124	Cron	New	This is an application development server
LINUX05	lat125.pbm.ihost.com	129.40.178.125	Cron	New	Test server with LDAD, automount /home/
LINUX06	lat126.pbm.ihost.com	129.40.178.126	Cron	New	This is a second Web Server

http://129.40.178.128/cgi-bin/getVMdata.sh?LINUX01

Figure 13-8 Linux report

Linux report drill down pages

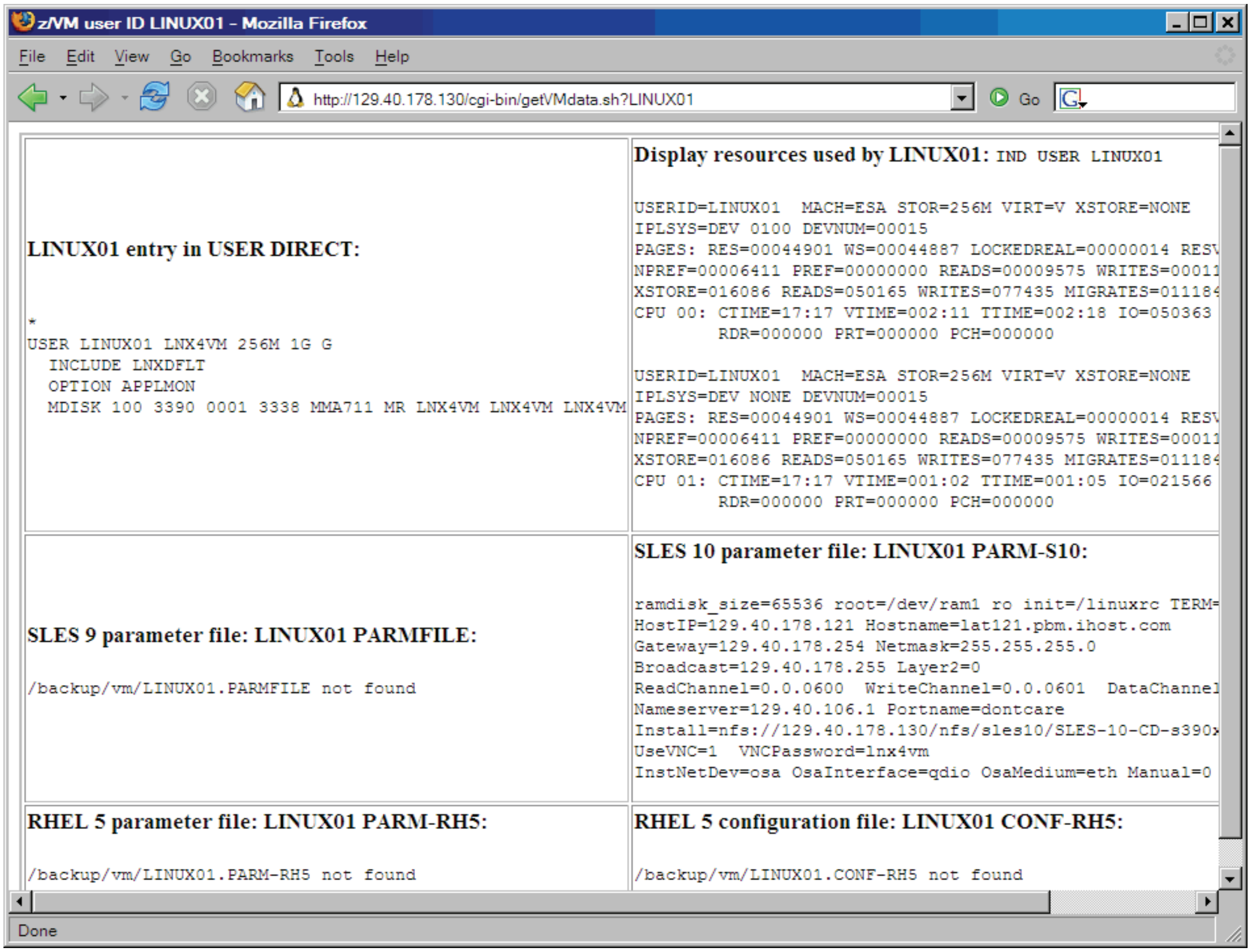
From the Linux report page, you can drill down into the following columns:

z/VM User ID	z/VM-related data
Linux host name	Linux non-networking related data
IP address	Linux networking related data
Sitar data	Output of Sitar - either from the nightly cron run, or a new report in real time

In the right-most column, **Description/Log**, you can keep a record of each virtual server. The data is retrieved from and stored in the file `descLog.txt` in each of the virtual server's directories under `/backup/linux/`. These files can be viewed and edited directly from the Web application.

z/VM User ID drill down

Following is a screen capture of the z/VM related data when drilling down under the **z/VM user ID** column:


z/VM user ID LINUX01 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.130/cgi-bin/getVMdata.sh?LINUX01
Go
Done

LINUX01 entry in USER DIRECT:

```
*
USER LINUX01 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3338 MMA711 MR LNX4VM LNX4VM LNX4VM
```

Display resources used by LINUX01: `IND USER LINUX01`

```
USERID=LINUX01  MACH=ESA STOR=256M VIRT=V XSTORE=NONE
IPLSYS=DEV 0100 DEVNUM=00015
PAGES: RES=00044901 WS=00044887 LOCKEDREAL=00000014 RESV
NPREF=00006411 PREF=00000000 READS=00009575 WRITES=00011
XSTORE=016086 READS=050165 WRITES=077435 MIGRATES=011184
CPU 00: CTIME=17:17 VTIME=002:11 TTIME=002:18 IO=050363
        RDR=000000 PRT=000000 PCH=000000

USERID=LINUX01  MACH=ESA STOR=256M VIRT=V XSTORE=NONE
IPLSYS=DEV NONE DEVNUM=00015
PAGES: RES=00044901 WS=00044887 LOCKEDREAL=00000014 RESV
NPREF=00006411 PREF=00000000 READS=00009575 WRITES=00011
XSTORE=016086 READS=050165 WRITES=077435 MIGRATES=011184
CPU 01: CTIME=17:17 VTIME=001:02 TTIME=001:05 IO=021566
        RDR=000000 PRT=000000 PCH=000000
```

SLES 9 parameter file: LINUX01 PARMFILE:

```
/backup/vm/LINUX01.PARMFILE not found
```

SLES 10 parameter file: LINUX01 PARM-S10:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=
HostIP=129.40.178.121 Hostname=lat121.pbm.ihost.com
Gateway=129.40.178.254 Netmask=255.255.255.0
Broadcast=129.40.178.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel
Nameserver=129.40.106.1 Portname=dontcare
Install=nfs://129.40.178.130/nfs/sles10/SLES-10-CD-s390x
UseVNC=1  VNCPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

RHEL 5 parameter file: LINUX01 PARM-RH5:

```
/backup/vm/LINUX01.PARM-RH5 not found
```

RHEL 5 configuration file: LINUX01 CONF-RH5:

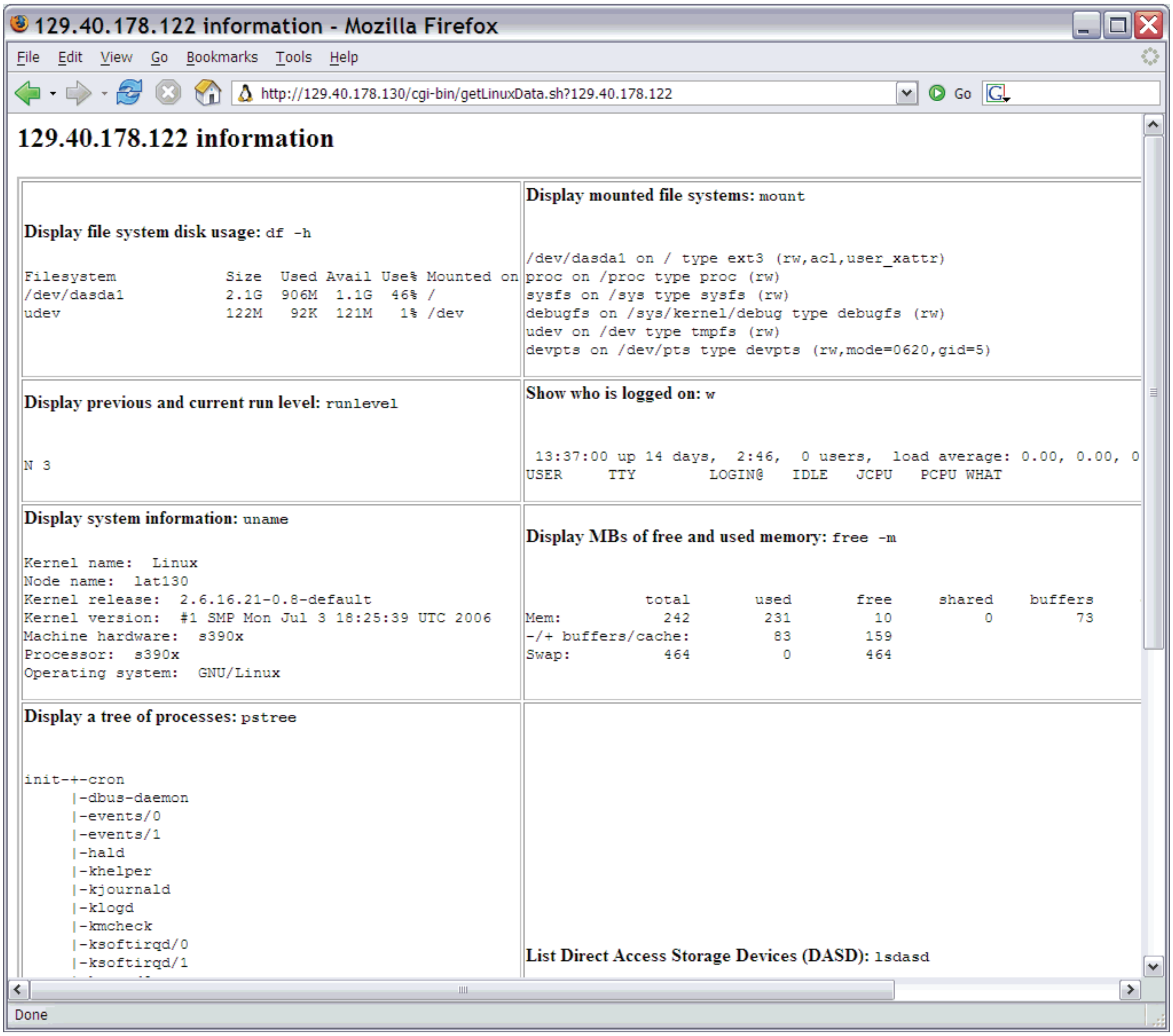
```
/backup/vm/LINUX01.CONF-RH5 not found
```

Figure 13-9 z/VM user ID drill down report

The first table cell in the upper left corner shows the directory entry of the specified user ID extracted from the `USER DIRECT` file. This file must be up to date on the controller. There is a checkbox, **Refresh /backup/vm**, on the DAZL home page that allows you to update all z/VM configuration files including the `USER DIRECT` file. The table cell in the upper right shows the output of the **`INDICATE USER`** command for the specified user ID. The bottom two table cells display the contents of the installation parameter files with file types of `PARMFILE` for SLES 9 and `PARM-S10` for SLES 10, or `PARM-RH5` and `CONF-RH5` for RHEL5.

Linux host name drill down

Following is a screen capture of the z/VM related data when drilling down under the **Linux host name** column:


129.40.178.122 information - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.130/cgi-bin/getLinuxData.sh?129.40.178.122
Go
Done

129.40.178.122 information

Display file system disk usage: `df -h`

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.1G  906M  1.1G  46% /
udev                  122M   92K  121M   1% /dev
```

Display mounted file systems: `mount`

```
/dev/dasda1 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
```

Display previous and current run level: `runlevel`

```
N 3
```

Show who is logged on: `w`

```
 13:37:00 up 14 days,  2:46,  0 users,  load average: 0.00, 0.00, 0
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
```

Display system information: `uname`

```
Kernel name:  Linux
Node name:  lat130
Kernel release:  2.6.16.21-0.8-default
Kernel version:  #1 SMP Mon Jul 3 18:25:39 UTC 2006
Machine hardware:  s390x
Processor:  s390x
Operating system:  GNU/Linux
```

Display MBs of free and used memory: `free -m`

```
             total       used       free     shared    buffers
Mem:           242        231         10          0         73
-/+ buffers/cache:         83        159
Swap:          464          0        464
```

Display a tree of processes: `pstree`

```
init-+-cron
     |-dbus-daemon
     |-events/0
     |-events/1
     |-hald
     |-khelper
     |-kjournald
     |-klogd
     |-kmcheck
     |-ksoftirqd/0
     |-ksoftirqd/1
```

List Direct Access Storage Devices (DASD): `lsdasd`

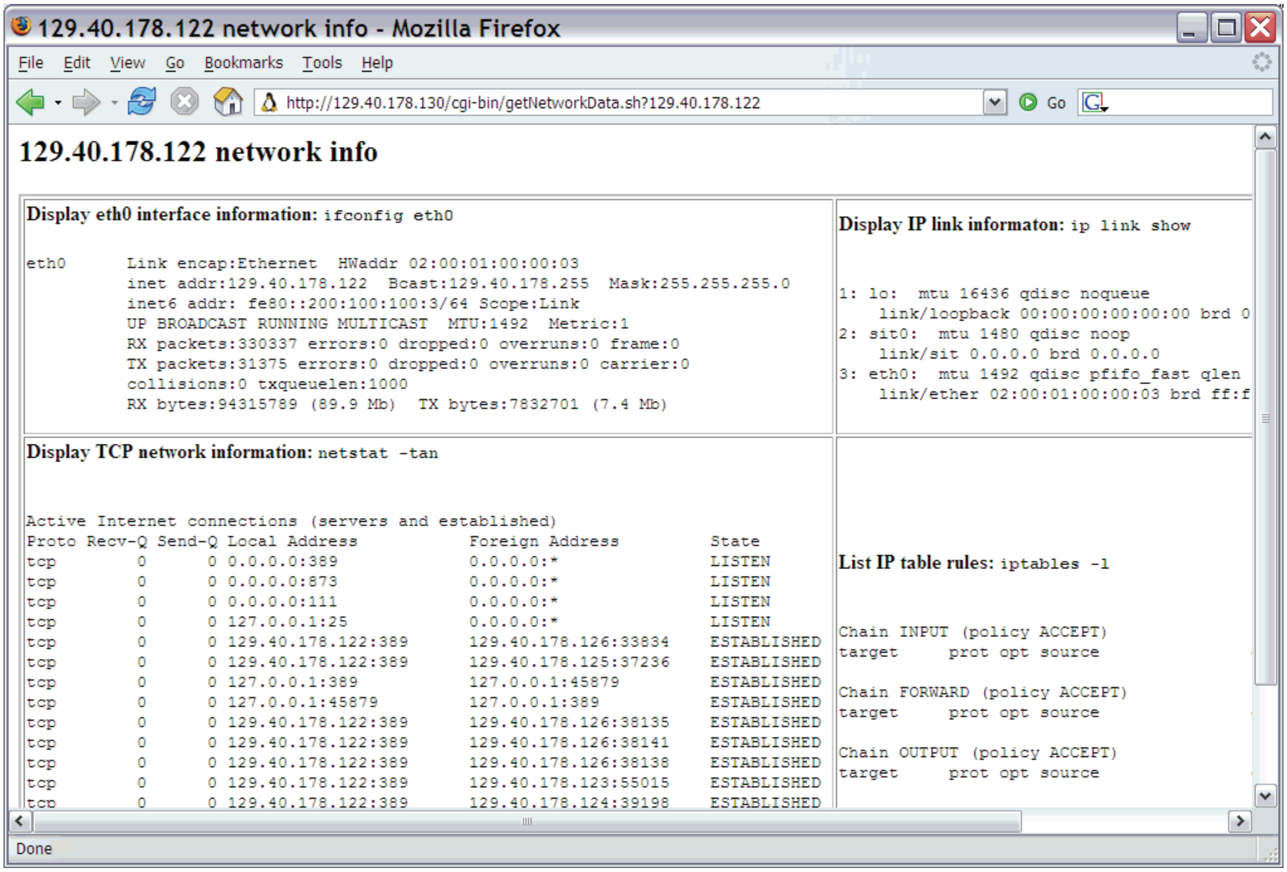
Figure 13-10 Linux information drill down report

This report gives the output from the following commonly run commands to query the status of Linux systems.

`df -h`	Display file system disk usage
`mount`	Display mounted file systems
`runlevel`	Display previous and current run levels
`w`	Show who is logged on
`uname`	Display system information about this distribution
`free -m`	Display megabytes of free and used memory
`pstree`	Display a tree of processes
`lsdasd`	List DASD (disks)

IP address drill down

Following is a screen capture of the z/VM related data when drilling down under the **IP address** column:


129.40.178.122 network info - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
http://129.40.178.130/cgi-bin/getNetworkData.sh?129.40.178.122
Go
Done

129.40.178.122 network info

Display eth0 interface information: `ifconfig eth0`

```
eth0      Link encap:Ethernet  HWaddr 02:00:01:00:00:03
          inet addr:129.40.178.122  Bcast:129.40.178.255  Mask:255.255.255.0
          inet6 addr: fe80::200:100:100:3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
          RX packets:330337 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31375 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:94315789 (89.9 Mb)  TX bytes:7832701 (7.4 Mb)
```

Display IP link informaton: `ip link show`

```
1: lo:  mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 0
2: sit0:  mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
3: eth0:  mtu 1492 qdisc pfifo_fast qlen
    link/ether 02:00:01:00:00:03 brd ff:f
```

Display TCP network information: `netstat -tan`

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:873             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp        0      0 129.40.178.122:389      129.40.178.126:33834    ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.125:37236    ESTABLISHED
tcp        0      0 127.0.0.1:389           127.0.0.1:45879         ESTABLISHED
tcp        0      0 127.0.0.1:45879         127.0.0.1:389           ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.126:38135    ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.126:38141    ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.126:38138    ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.123:55015    ESTABLISHED
tcp        0      0 129.40.178.122:389      129.40.178.124:39198    ESTABLISHED
```

List IP table rules: `iptables -l`

```
Chain INPUT (policy ACCEPT)
target     prot opt source

Chain FORWARD (policy ACCEPT)
target     prot opt source

Chain OUTPUT (policy ACCEPT)
target     prot opt source
```

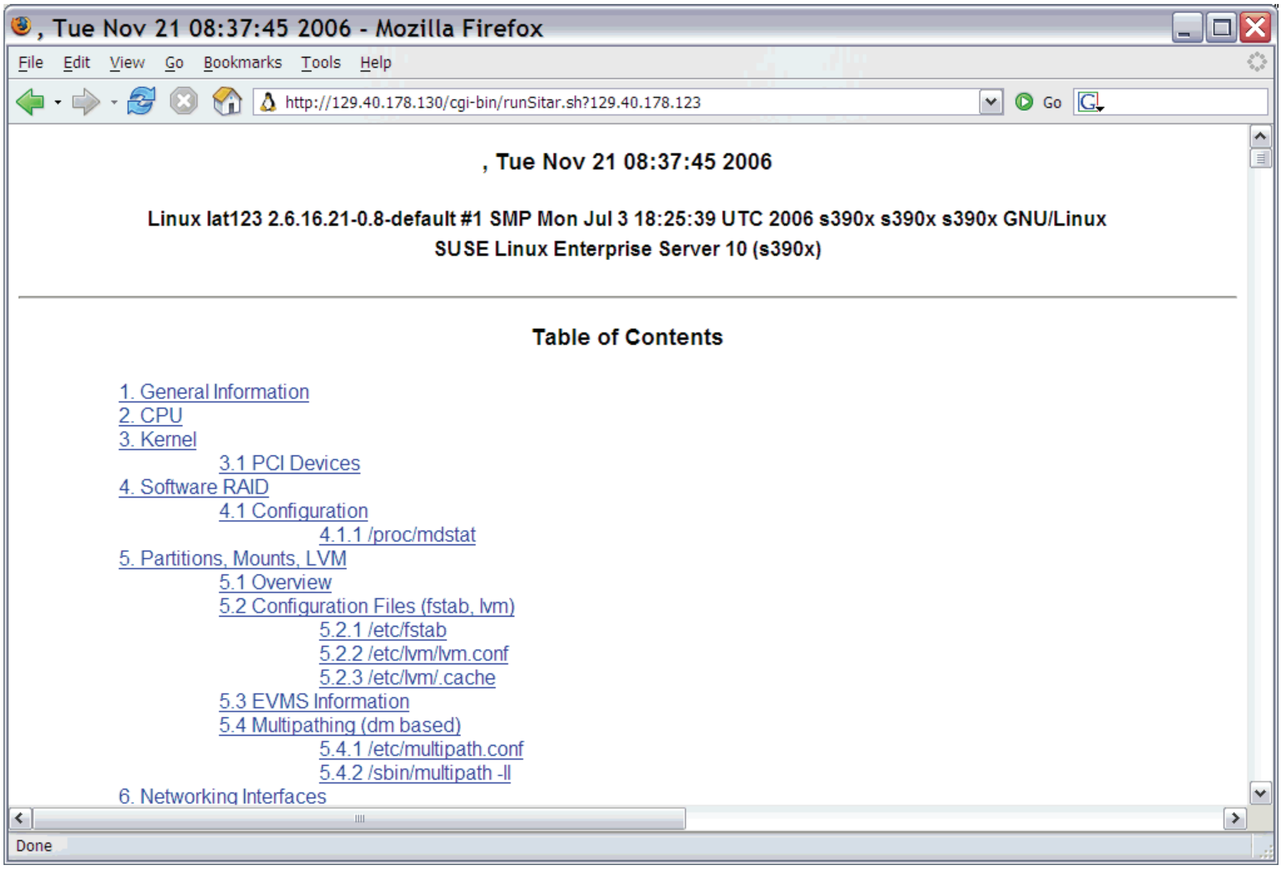
Figure 13-11 Linux networking drill down report

This report is generated from the output of the following four network-related commands:

`ifconfig eth0`	Display information about the interface eth0
`ip link show`	Display IP link information
`netstat -tan`	Display TCP network information
`iptables -L`	List IP tables (firewall) rules

Sitar data drill down

You can drill down into the **Cron** link which will use the Sitar output file that should be generated by a cron job each night. Also you can drill down into the **New** link which will generate a Sitar HTML file on demand. Following is a screen capture of the z/VM related data when drilling down under the **Sitar** column:



, Tue Nov 21 08:37:45 2006 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.130/cgi-bin/runSitar.sh?129.40.178.123
Go
Done

, Tue Nov 21 08:37:45 2006

Linux lat123 2.6.16.21-0.8-default #1 SMP Mon Jul 3 18:25:39 UTC 2006 s390x s390x s390x GNU/Linux
SUSE Linux Enterprise Server 10 (s390x)

Table of Contents

1. General Information
2. CPU
3. Kernel
    3.1 PCI Devices
4. Software RAID
    4.1 Configuration
        4.1.1 /proc/mdstat
5. Partitions, Mounts, LVM
    5.1 Overview
    5.2 Configuration Files (fstab, lvm)
        5.2.1 /etc/fstab
        5.2.2 /etc/lvm/lvm.conf
        5.2.3 /etc/lvm/.cache
    5.3 EVMS Information
    5.4 Multipathing (dm based)
        5.4.1 /etc/multipath.conf
        5.4.2 /sbin/multipath -ll
6. Networking Interfaces

Figure 13-12 Sitar drill down report

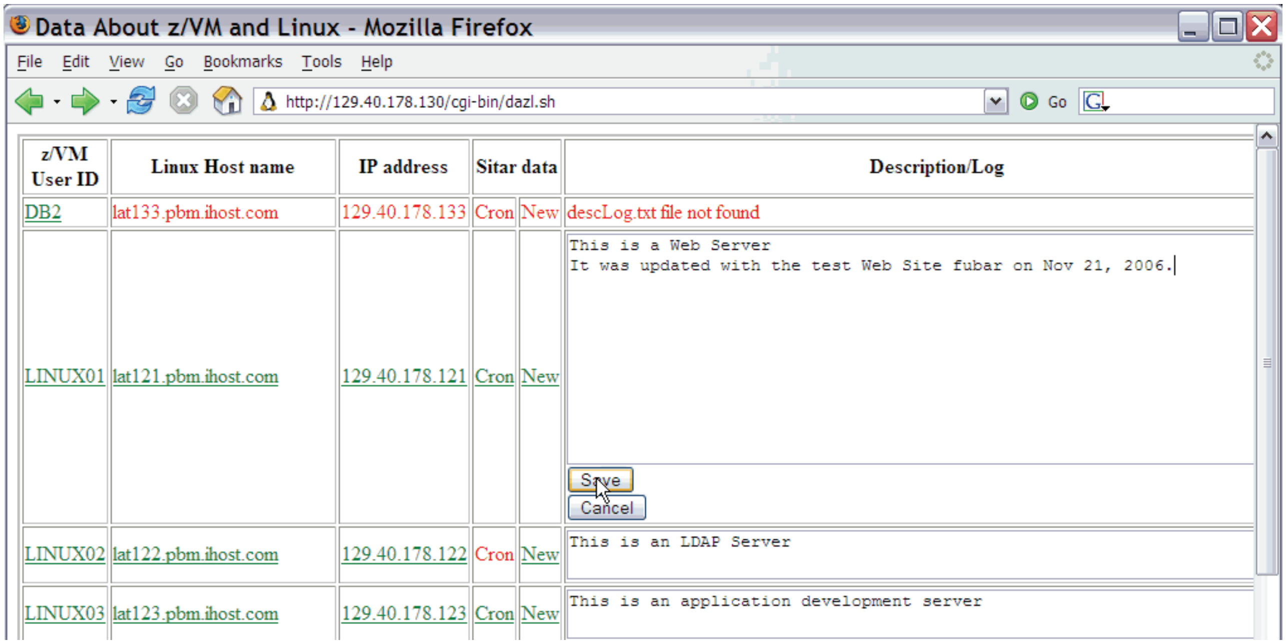
Linux report log file editing

Another useful feature of this Web application is the ability to maintain a flat text file for each of the Linux virtual servers from a browser. The basic premise is that the first line of the file is an overall description of the virtual server and the remainder of the file can be used to log changes. This feature utilizes Asynchronous Java and XML (AJAX) for pushing and pulling changes between the browser and the Web server without having to update the entire page.

The flat text files are maintained on the controller under the directory `/backup/linux/` with one file named `descLog.txt` under the each Linux subdirectory (of the form <userID>-on-<TCP.IP.address>. For example, a Linux system running on the user ID `LINUX01` with a TCP/IP address of 129.40.178.121 would have a flat text file named `/backup/linux/LINUX01-on-129.40.178.121/descLog.txt`

You can these files manually via an SSH session on the controller. That is still true, but editing the files via a Web interface is another, perhaps more convenient, way of viewing and editing a set of files. Please note there is no mechanism for locking files. If you are editing a file via the Web interface and in the file system (or from two different browsers), then it is possible to lose changes. Adding a locking mechanism is left as an exercise for the reader :)).

Following is a screen capture of one Linux description/log file being edited:



Data About z/VM and Linux - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://129.40.178.130/cgi-bin/dazl.sh
Go
Save
Cancel

z/VM User ID	Linux Host name	IP address	Sitar data		Description/Log
DB2	lat133.pbm.ihost.com	129.40.178.133	Cron	New	descLog.txt file not found
LINUX01	lat121.pbm.ihost.com	129.40.178.121	Cron	New	This is a Web Server It was updated with the test Web Site fubar on Nov 21, 2006.
LINUX02	lat122.pbm.ihost.com	129.40.178.122	Cron	New	This is an LDAP Server
LINUX03	lat123.pbm.ihost.com	129.40.178.123	Cron	New	This is an application development server

Figure 13-13 Editing Linux description/log files in place

13.5.3 DAZL design

This section discusses some of the design of DAZL.

The main page, `index.html`, is a static HTML page which invokes one of three Common Gateway Interface (CGI) bash scripts. These scripts are run on the server in real time to generate reports. Those scripts in turn, invoke other CGI scripts where the reports have drill down links. The ability to edit description/log files in place utilizes Java Script and AJAX.

The xhr class and motivation to use AJAX was obtained from the book *Build Your Own AJAX Web Applications*, by Matthew Eernisse, published by SitePoint Ltd in 2006. See the following Web site for details on this book:

http://www.sitepoint.com/books/ajax1/

The Java Script file, xhr.js, is included, unchanged, with permission. It is a part of the Fleegix.js JavaScript Toolkit, copyrighted and distributed under the Apache License, v2.0. It is available from:

http://js.fleegix.org/

The main page is named `index.html` which goes in Apache's DocumentRoot, `/srv/www/htdocs/`, by default in SLES 9 and 10 or `/var/www/html` in RHEL 5. In addition there is the AJAX Java Script file `xhr.js`, obtained from the book above, and the Java Script `editLog.js` written to edit the text files. The CGI scripts reside in Apache's CGI directory, `/srv/www/cgi-bin/`, by default in SLES 9 and 10. They are as follow:

`dazd.sh`	Create the DASD report from the `USER DISKMAP` file
`dazl.sh`	Create the Linux report from the directories under `/backup/linux/`
`dazvm.sh`	Create the z/VM status report
`getDasdInfo.sh`	Create a simple page from z/VM **`QUERY DASD DETAILS`** command
`getDesc.sh`	Return the first line (the description) of the file `descLog.txt`

getLinuxData.sh Create a table page from commonly used Linux commands
getLog.sh Return the entire file `descLog.txt`
getNetworkData.sh Create a table page from commonly used Linux network commands
indQ.sh Create a simple page from z/VM **QUERY DASD DETAILS** command
qVswitch.sh Create a simple page from the z/VM **QUERY VSWITCH** command
getVMdata.sh Create a page of z/VM data related to one Linux virtual server
runSitar.sh Create a Sitar HTML file for a Linux virtual server on demand
getVolMap.sh Create a page with value from USER DISKMAP for a z/VM volume
saveLog.sh Save in-place changes to descLog.txt file for a Linux virtual server

Following is a block diagram of the DAZL system from a programming point of view:

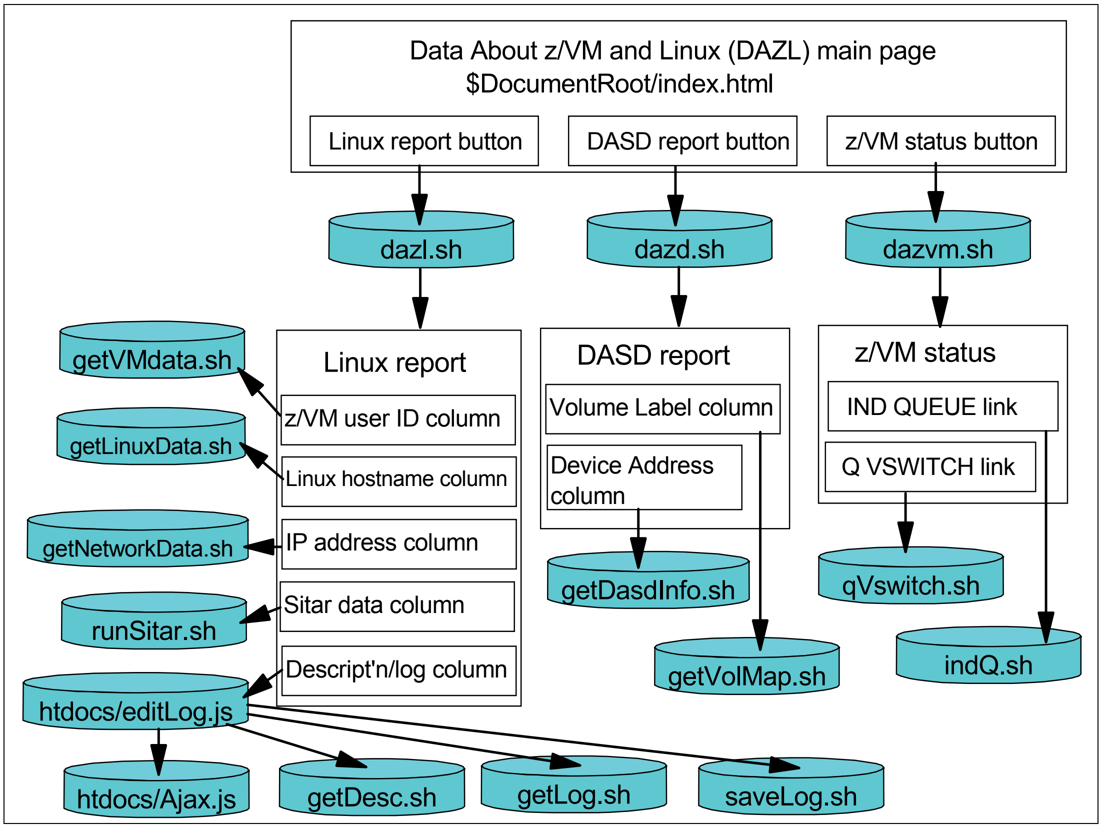


Figure 13-14 DAZL design block diagram

14

Backup and restore

I never think of the future. It comes soon enough.

--Albert Einstein

Given the opening quote, Albert Einstein would probably have not have made a good z/VM and Linux system administrator :)).

Backup and restore can be divided into two fundamental categories, each of which must be able to answer a basic question:

Incremental back up	How do I quickly get back a file that was deleted or corrupted?
Disaster recovery	How do I restore my systems if my entire data center is destroyed?

A key attribute of incremental back up is self-service. Ideally, you as the system administrator should not have to restore individual files that have been deleted or corrupted, rather, you should be easily be able to tell end users how to do it themselves. Or at least you should be able to get access to the files quickly.

A key to disaster recovery is practice. There is no *silver bullet*; disaster recovery takes planning and hard work. One rule of disaster recovery to consider is “For every backup you make, you should do one restore”. This chapter only touches on disaster recovery in the final section.

Getting back to incremental backup and restore, both your z/VM system and your Linux virtual servers must be considered. The following sections address this:

- “Incremental backup of z/VM” on page 224
- “Incremental backup of Linux virtual servers” on page 227

14.1 Incremental backup of z/VM

The following configuration files on z/VM are important:

- SYSTEM CONFIG on the MAINT CF1 minidisk
- USER DIRECT on the MAINT 2CC minidisk
- <system_ID> TCPIP and SYSTEM DTCPARMS on the TCPMAINT 198 minidisk
- TCPIP DATA on the TCPMAINT 592 minidisk
- PROFILE EXEC on the AUTOLOG1 191 minidisk

In addition, it would be helpful to backup all files on the LNXMAINT 192 minidisk. Two bash scripts are written to back up these files and other z/VM data:

backupVM.sh Backs up the files listed above. This script should not be your primary backup procedure, but it may help as a secondary method.

getVMinfo.sh Runs some basic CP commands to capture information that may be helpful.

14.1.1 The getVMinfo.sh script

There are some CP commands which also capture important information about the systems. A small script, **getVMinfo.sh** is supplied to issue the CP commands. This file should have been copied to the directory /usr/local/sbin/:

```
# cat /usr/local/sbin/getVMinfo
#!/bin/bash
...
echo "output of CP QUERY PROC:"
echo "-----------------------"
vmcp QUERY PROC
echo ""
echo "output of CP QUERY FILES:"
echo "------------------------"
vmcp QUERY FILES
echo ""
echo "output of CP QUERY NSS ALL MAP:"
echo "------------------------------"
vmcp QUERY NSS ALL MAP
echo ""
echo "output of CP QUERY CPLEVEL:"
echo "--------------------------"
vmcp QUERY CPLEVEL
```

Running getVMinfo.sh script

Try running the script. You should see output similar to the following:

```
# getVMinfo.sh
output of CP QUERY PROC:
-----------------------
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
...

output of CP QUERY FILES:
------------------------
FILES: 0000133 RDR, 0000018 PRT,      NO PUN

output of CP QUERY NSS ALL MAP:
------------------------------
```

```
FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0043 CMS      NSS      0000256K 00000  0000D   EW  A  00005   00-15     NO
...
output of CP QUERY CPLEVEL:
---------------------------
z/VM Version 5 Release 2.0, service level 0601 (64-bit)
Generated at 02/13/06 11:45:25 EDT
IPL at 08/14/06 08:51:11 EDT
```

14.1.2 The backupVM.sh script

The **backupVM.sh** script backs up important z/VM files to the /backup/vm/ directory. The first command that it runs is the above **getVMinfo.sh** script.

Important: Please note two important aspects of this script

1. It copies the USER DIRECT file which contains all of the system's passwords (unless you are also using an external security manager). So security to the controller becomes as important as security to the MAINT user ID.
2. It backs up important z/VM data to a Linux system running under the same z/VM. If your z/VM system cannot be brought back, then this Linux system cannot be started and you cannot get the important z/VM data. Therefore, you may consider copying the output of this script to a different LPAR or different physical server.

Following is the main portion of the **backupVM.sh** script:

```
# cd /usr/local/sbin/
# cat backupVM.sh
...
# main()
cd /backup/vm              # change directory to /backup/vm where data is saved
getVMinfo.sh > zVMinfo.txt # run getVMinfo.sh to save basic VM information

# get USER DISKMAP from MAINT 191
linkAccess MAINT 191 as 1191
getFile USER.DISKMAP from 1191
relDetach 1191

# get USER DIRECT from MAINT 2CC
linkAccess MAINT 2CC as 12cc
getFile USER.DIRECT from 12cc
relDetach 12cc

# get SYSTEM CONFIG from MAINT CF1
linkAccess MAINT CF1 as 1cf1
getFile SYSTEM.CONFIG from 1cf1
relDetach 1cf1

# get TCPIP files from TCPMAINT 198
systemID=`vmcp q userid | awk '{print $3}'`
linkAccess TCPMAINT 198 as 1198
getFile $systemID.TCPIP from 1198
getFile SYSTEM.DTCPARMS from 1198
relDetach 1198
```

```
# get TCPIP DATA from TCPMAINT 592
linkAccess TCPMAINT 592 as 1592
getFile TCPIP.DATA from 1592
relDetach 1592

# get PROFILE EXEC from AUTOLOG1 191 - rename to different FT
linkAccess AUTOLOG1 191 as 2191
CMSdevice=`cat /proc/dasd/devices | grep 2191 | awk '{print $7}'`
cmsfscat -a -d /dev/$CMSdevice PROFILE.EXEC > PROFILE.EXECAUT1
relDetach 2191

getFiles 191        # Get all files from LNXMAINT 192 which is our 191 disk
```

The `linkAccess()` function uses the CP **LINK** command via **vmcp** to link to the specified disk read-only, then the **chccwdev** command to enable the new disk. The `getFile()` function uses the **cmsfslst** command to copy a single file to the `/backup/vm/` directory.

Running backupVM.sh script

Try running it now. You may choose to first remove any existing files. You should see output similar to the following:

```
# cd /backup/vm
# rm *          // be sure you are in the right directory when you issue this!
# backupVM.sh
Backing up USER.DISKMAP seems to have succeeded
Backing up USER.DIRECT seems to have succeeded
Backing up SYSTEM.CONFIG seems to have succeeded
Backing up LNXVM52.TCPIP seems to have succeeded
Backing up SYSTEM.DTCPARMS seems to have succeeded
Backing up TCPIP.DATA seems to have succeeded
Backing up CHPW52.XEDIT seems to have succeeded
...
Backing up SLES-S10.EXEC seems to have succeeded
```

You should see a number of files copied. All text files are converted to ASCII so you should be able to read them from Linux. Your output will differ:

```
# ls -F
CHPW52.XEDIT      LINUX04.PARMFILE  SLES-S10.EXEC    SYSTEM.CONFIG
CPFORMAT.EXEC     LINUX05.PARMFILE  SLES10.EXEC      SYSTEM.DTCPARMS
LABEL520.EXEC     LINUX06.PARMFILE  SLES10.INITRD    TCPIP.DATA
LABEL520.XEDIT    LINUX07.PARMFILE  SLES10.KERNEL    USER.DIRECT
LINUX01.PARMFILE  LNXVM52.TCPIP     SLES10.PARM-S10  USER.DISKMAP
LINUX02.PARMFILE  PROFILE.EXEC      SLES10.PARMFILE  zVMinfo.txt
LINUX03.PARMFILE  PROFILE.EXECAUT1  SWAPGEN.EXEC
```

Running backupVM.sh daily via cron

The **backupVM.sh** script is run once a day on the controller via the **cron** daemon from the script **/etc/cron.daily/nightly.sh**:

```
# tail -4 /etc/cron.daily/nightly.sh
# run sitar, backupVM.sh and backupLinux.sh nightly on the controller
sitar --format=html --outfile=/srv/www/htdocs/sitar.html
backupVM.sh            # backup z/VM files to /backup/vm/
backupLinux.sh         # backup Linux files to /backup/linux/
```

If this does successfully run each night, you will always have backed up files that are no older than one day.

14.2 Incremental backup of Linux virtual servers

Your IT shop may have an existing backup/restore solution for Linux. For example:

- Computer Associates' Brightstor Enterprise Backup
- Innovation Data Processing's FDR/UPSTREAM
- IBM's Tivioli Storage Manager (TSM)
- SecureAgent Software's SecureBackup
- Veritas' Backup Exec

For details of these and other solutions, see the *Software Developer Products for Linux on IBM System z and S/390* page on the Web at:

http://www-1.ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html

Descriptions of these products are outside the scope of this book, and they are a much more complete solution than what is described in the sections that follow.

Some simple **cron** scripts and configuration of **rsync** follow. These will result in the /etc/ directories and other information of the virtual servers to be backed up to the controller nightly.

14.2.1 Configuring rsync on the controller

Because the master image was set up to allow the controller to copy its /etc/ directory and for ssh key-based authentication, each of the virtual servers' /etc/ directories can be copied with a single **rsync** command. For example, the following commands will copy the /etc/ directory on LINUX01 at IP address 129.40.178.121:

```
# cd /backup/LINUX01-on-129.40.178.121
# rsync -r --timeout=30 129.40.178.121:/etc .
```

To perform this backup on each of the virtual servers, a script named **backupLinux.sh** is provided. It iterates through the directories created by the clone script under /backup/linux/ of the form <userID>-on-<IPaddress>:

```
# cd /usr/local/sbin
# cat backupLinux.sh
#!/bin/bash
...
BACKUP="/backup/linux"
cd $BACKUP
for i in *-on-*            # iterate through dirs with "-on-" in their names
do
  IP_addr=${i#*-on-}       # this chops the head off and grabs the IP address
  cd $BACKUP/$i            # change directory to next virtual server
  rsync -r --timeout=3 $IP_addr:/etc . > /dev/null 2>&1 # use rsync to backup
  rc=$?
  if [ $rc != 0 ]; then    # error
    echo "Error: rsync returned $rc while backing up $i"
  else
    echo "Backing up $i seems to have succeeded"
  fi
done
```

This script should backup the /etc/ directory of each Linux system cloned to the appropriate directory under /backup/linux/.

14.2.2 Configuring sitar to run nightly on the virtual servers

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

This command can easily be run nightly via **cron**:

```
# cd /etc/cron.daily
# vi run-sitar    // add two lines
#!/bin/bash
sitar --format=html --outfile=/etc/sitar.html
```

14.3 Disaster recovery of z/VM and virtual Linux servers

In addition to incremental backups, you should do regular system backups of your z/VM system. Sections 4.10, “Backing up your z/VM system to tape” on page 61 and 4.12, “Restoring your z/VM system from tape” on page 67 just touched on this.

You may choose to do a full volume backups of all DASD in the LPAR. Then in the event of a disaster, your entire system could be restored. This is the most thorough and best way to perform backups, and is one of the areas where the System z platform excels.

You may also choose to backup a subset of the volumes (or perhaps you may choose to backup all volumes once a quarter, but backup a subset once a month). Two critical user IDs are LNXMAINT and the controller user ID. If you have followed all steps in this book, then each of the virtual servers /etc/ file system is being backed to /backup/ file system which is the controller user ID’s 203 minidisk in this book. The master image root file system (100 minidisk) and the controller root file system (200 minidisk) are also critical. Given that these minidisks are vital to your system, you may choose to back them up more regularly or in a different fashion. You can look at the USER DISKMAP file, created on the **MAINT 191** disk by the **DISKMAP** command, to determine which volumes to back up. You will probably have many more critical volumes than these three, but consider these especially in your hierarchy of backed up data.

A

References

This book refers to additional material that can be downloaded from the Internet as described below.

A.1 Related books

These publications are also relevant as further information sources:

- *Linux for zSeries and S/390 Device Drivers, Features, and Commands*, LNUX-1403
- *SUSE LINUX Enterprise Server: INSTALLATION AND ADMINISTRATION*
- *SUSE LINUX Enterprise Server: ARCHITECTURE-SPECIFIC INFORMATION*
- *SUSE LINUX Enterprise Server: START-UP GUIDE*
- *z/VM Guide for Automated Installation and Service: Version 5 Release 1.0*, GC24-6099
- *z/VM System Messages and Codes — CP: Version 5 Release 1.0*, GC24-6119
- *z/VM TCP/IP Messages and Codes: Version 5 Release 1.0*, GC24-6124
- *The Program Directory for Performance Toolkit for VM*, GI11-4800
- *z/VM CP Commands and Utilities Reference: Version 5 Release 1.0*, SC24-6081
- *z/VM CP Planning and Administration: Version 5 Release 1.0*, SC24-6083
- *z/VM Getting Started with Linux on zSeries: Version 5 Release 1.0*, SC24-6096
- *z/VM TCP/IP Planning and Customization: Version 5 Release 1.0*, SC24-6125
- *z/VM Performance Toolkit*, SC24-6136
- *Communication Controller for Linux on zSeries V1.0 Implementation Guide*, SC31-6872
- *Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864

A.2 Online resources

These Web sites and URLs are also relevant as further information sources:

- The Linux for zSeries and S/390 portal:

  http://linuxvm.org/

- The linux-390 list server:

  http://www2.marist.edu/htbin/wlvindex?linux-390

- Linux 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/

A.3 Important z/VM files

z/VM differs from Linux in regard to the location and number of configuration files. In Linux, there are many configuration files and most of them are in or under the /etc/ directory. On z/VM, there are relatively few configuration files. However, they are on many different minidisks. Table 14-1provides a summary and the location of important z/VM configuration files.

Table 14-1 Important z/VM configuration files

File	Location	Description
SYSTEM CONFIG	MAINT CF1	This is the operating system's main configuration file. It defines the system name, the CP volumes, User volumes and other settings.
USER DIRECT	MAINT 2CC	This file defines the user directory. All user IDs or virtual machines known to the system are defined here (assuming a directory maintenance product is not being used).
<System_ID> TCPIP	TCPMAINT 198	This file defines the resources for the primary z/VM TCP/IP stack, including TCP/IP address, OSA resources, subnet mask and gateway. It is initially created by the IPWIZARD tool as `PROFILE TCPIP`.
SYSTEM DTCPARMS	TCPMAINT 198	This file is created to define the TCP/IP stacks on the system. It is initially created by the IPWIZARD tool.
TCPIP DATA	TCPMAINT 592	This file defines the DNS server, the domain name and some other settings. It is initially created by the IPWIZARD tool.
PROFILE EXEC	AUTOLOG1 191	This file is a REXX EXEC that is run when the system starts up. It is analogous to the /etc/inittab file in Linux.

A.4 Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors

A.4.1 XEDIT cheat sheet

XEDIT has line commands which are typed on the command line (===>) and prefix commands which are typed over the line numbers on the left side of the screen.

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.4.2 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

Common source code

This section lists source code for z/VM and Linux that is common regardless of Linux distribution. The common code is in the following categories

- z/VM REXX EXECs and XEDIT macros
- Bash scripts on the master image
- Bash scripts on the controller

B.1 z/VM REXX EXECs and XEDIT macros

This section lists z/VM REXX EXECs and XEDIT macros.

B.1.1 The CPFORMAT EXEC

Following is the code for the EXEC that formats multiple disks via **CPFMTXA** (described in section 4.6.1, “Formatting the paging volumes” on page 47):

```
/*+------------------------------------------------------------------+*/
/*| EXEC: CPFORMAT - wrapper around CPFMTXA to format many DASD      |*/
/*|  retVal: 0 - success                                             |*/
/*|          1 - help was asked for or given                         |*/
/*|          2 - user is not sure                                    |*/
/*|          3 - DASD (minidisk) range is not valid                  |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT   |*/
/*+------------------------------------------------------------------+*/

/*--------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
```

```
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
--------------------------------------------------------------------*/

firstChar = 'M' /* change this for an LPAR ID other than 'M' */
parse upper arg dasds "AS " type
if ((dasds = '') | (dasds = '?')) then call help
labelPrefix = getLabelPrefix(firstChar type)
numDasd = parseDasd(dasds)
answer = areYouSure(type)
if (answer = 'Y') then  /* the user is sure */
do
  retVal = doFormat(labelPrefix numDasd type)
  call doReport
end
else
  retVal = 2
exit retVal

/*+-------------------------------------------------------------------+*/
help: procedure
/*+-------------------------------------------------------------------+*/
  parse source . . fn .
  say ''
  say 'Synopsis:'
  say ''
  say '  Format one or a range of DASD as page, perm, spool or temp disk space'
  say '  The label written to each DASD is V<t><xxxx> where:'
  say '    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)'
  say '    <xxxx> is the 4 digit address'
  say ''
  say 'Syntax is:'
  say "                                              .-PAGE-."
  say "   >>--CPFORMAT--.-rdev---------------.--AS---+-PERM-+---------><"
  say "                 | <----------------< |       '-SPOL-'"
  say "                 '-rdev1-rdev2--------'  "
  say ''
exit 1

/*+-------------------------------------------------------------------+*/
areYouSure: procedure
/*| Show minidisks, ask are you sure                                  |*/
/*|  parm 1: type - PERM, PAGE, or SPOL                               |*/
/*|  retVal: firstChar - LPAR identifier, 'V' by default              |*/
/*+-------------------------------------------------------------------+*/
  arg type
  say ''
  say 'WARNING - this will destroy data!'
  say 'ARE YOU SURE you want to format the DASD as' type 'space (y/n)?'
  parse upper pull answer
return  substr(answer, 1, 1) /* from areYouSure */

/*+-------------------------------------------------------------------+*/
getLabelPrefix: procedure
/*| Return first two chararcters of label                             |*/
/*|  parm 1: firstChar - LPAR identifier, 'V' by default              |*/
/*|  retVal: the two character label prefix                           |*/
/*+-------------------------------------------------------------------+*/
  arg firstChar type
  select
  when (type = PERM) then
```

```
    labelPrefix = firstChar||'M' /* for VM Minidisk */
  when (type = PAGE) then
    labelPrefix = firstChar||'P' /* for VM Page */
  when (type = SPOL) then
    labelPrefix = firstChar||'S' /* for VM Spool */
  otherwise
    do
      say 'Error: "AS" must be present and type must be PERM, PAGE or SPOL'
      call help
    end /* otherwise */
  end /* select */
return labelPrefix /* from getLabelPrefix */

/*+------------------------------------------------------------------+*/
parseDasd: procedure expose dasdList.
/*| parse all dasd into an array verifying all are attached          |*/
/*|  parm 1: dasds - the list of dasd passed in                      |*/
/*|  retVal: number of DASD in dasdList                              |*/
/*+------------------------------------------------------------------+*/
  arg dasds
  numDasd = 0
  say ''
  say 'Format the following DASD:'
  do while (dasds <> '')
    parse upper var dasds dasd dasds
    dashPos = pos('-', dasd)
    if (dashPos = 0) then  /* there is just one DASD */
    do
      numDasd = numDasd + 1
      dasdList.numDasd = dasd
      'CP Q MDISK' dasdList.numDasd 'LOCATION'
      if (rc <> 0) then
      do
        say 'Return code from Q MDISK =' rc
        say 'Are all DASD ATTached?'
        exit 3
      end
      call checkReserved(dasdList.numDasd)
    end /* do */
    else /* process the range of DASD */
    do
      startRange = substr(dasd, 1, dashPos - 1)
      endRange = substr(dasd, dashPos + 1, length(dasd) - dashPos)
      do i = x2d(startRange) to x2d(endRange)
        numDasd = numDasd + 1
        dasdList.numDasd = d2x(i)
        'CP Q MDISK' dasdList.numDasd 'LOCATION'
        if (rc <> 0) then
        do
          say 'Return code from Q MDISK =' rc
          exit 3
        end
        call checkReserved(dasdList.numDasd)
      end /* do i */
    end /* else */
  end /* do while */
return numDasd /* from parseDasd */

/*+------------------------------------------------------------------+*/
doFormat: procedure expose dasdList.
```

```
/*| Format all DASD specified using CPFMTXA                          |*/
/*|  parm 1: labelPrefix - the two character label prefix           |*/
/*|  parm 2: numDasd - number of DASD in the array dasdList          |*/
/*|  parm 3: type - the type of DASD format                          |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  arg labelPrefix numDasd type
  'CP TERM MORE 1 1'
  do i = 1 to numDasd
    label = getLabel(labelPrefix dasdList.i)
    call formatOne(dasdList.i type label)
  end /* do i = */
  'CP TERM MORE 50 10'
return 0 /* from doFormat */

/*+------------------------------------------------------------------+*/
checkReserved: procedure
/*| Try copying an already formatted DASD then relabelling it        |*/
/*|  parm 1: source                                                  |*/
/*|  parm 2: target                                                  |*/
/*|  parm 3: label                                                   |*/
/*+------------------------------------------------------------------+*/
  arg dasd
  /* create a list of reserved dasd - this is somewhat hokey to be sure
     but it's better to be hokey than to format system minidisks! */
  resvd1 = "0122 0123 0124 0125 0190 0191 0193 0194 019D 019E 0201 02A2"
  resvd2 = "02A4 02A6 02C2 02C4 02CC 02D2 0319 03A2 03A4 03A6 03B2 03C2"
  resvd3 = "03C4 03D2 0400 0401 0402 0405 0490 0493 049B 049E 04A2 04A4"
  resvd4 = "04A6 04B2 04C2 04C4 04D2 0500 051D 05A2 05A4 05A6 05B2 05C2"
  resvd5 = "05C4 05D2 05E5 05E6 06A2 06A4 06A6 06B2 06C2 06C4 06D2 07A2"
  resvd6 = "07A4 07A6 07B2 07C2 07C4 07D2 0CF1 0CF2 0CF3"
  reserved = resvd1 resvd2 resvd3 resvd4 resvd5 resvd6
  if (index(reserved, dasd) <> 0) then /* MAINT minidisk - ABORT! */
  do
    say 'Minidisk' dasd 'is a reserved MAINT minidisk'
    say 'This must be formatted manually using a different vaddr'
    exit 4
  end /* if dasd is reserved */
return /* from checkReserved */


/*+------------------------------------------------------------------+*/
doReport: procedure expose dasds
/*| Report on the newly labelled DASD                                |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  'DETACH' dasds
  'ATTACH' dasds '*'
  say ''
  say 'DASD status after:'
  'CP Q MDISK' dasds 'LOCATION'
return 0 /* from doReport */

/*+------------------------------------------------------------------+*/
formatOne: procedure
/*| Format a DASD via DDR                                            |*/
/*|  parm 1: disk - the vaddr to be formatted                        |*/
/*|  parm 2: type - PAGE, SPOL or PERM                               |*/
/*|  parm 3: label - the six character label                         |*/
/*+------------------------------------------------------------------+*/
```

```
  arg disk type label
  queue 'FORMAT'
  queue disk
  queue '0 END'
  queue label
  queue 'YES'
  queue type '0 END'
  queue 'END'
  'CPFMTXA'
return /* from formatOne */

/*+------------------------------------------------------------------+*/
getLabel: procedure
/*| Compose the six character label of a minidisk                    |*/
/*|  parm 1: labelPrefix - first two characters of label             |*/
/*|  parm 2: disk - vaddr of length 1, 2, 3 or 4                     |*/
/*|  return: the 6 character label                                   |*/
/*+------------------------------------------------------------------+*/
  arg labelPrefix disk
  diskLen = length(disk)
  select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
  end /* select */
return label /* from getLabel */
```

B.1.2 The CHPW52 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.2 USER DIRECT file:

```
/* CHPW52 XEDIT - change all passwords in z/VM 5.2 USER DIRECT file */

/*-------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
------------------------------------------------------------------*/

parse arg fn ft fm '(' options ')' newPass .
if (length(newPass) > 8) then
do
  say "Error: new password must be 8 characters or fewer"
  exit
end
say ''
say 'Changing all passwords to:' newPass
say ''

/* set some values */
'command set stay on'
'command set num on'
'command set nulls on'
'command set serial off'
'command set cmdline bottom'
'command set curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'

/* change user ID passwords */
'command c/CMS1  CMS1/CMS1' newPass'/*'
'command c/LGLOPR  LGLOPR/LGLOPR' newPass'/*'
'command c/MAINT MAINT/MAINT' newPass'/*'
'command c/CMSBATCH CMSBATCH/CMSBATCH' newPass'/*'
'command c/AVSVM AVSVM/AVSVM' newPass'/*'
'command c/TSAFVM TSAFVM/TSAFVM' newPass'/*'
'command c/VMSERVS VMSERVS/VMSERVS' newPass'/*'
'command c/VMSERVU VMSERVU/VMSERVU' newPass'/*'
'command c/VMSERVR VMSERVR/VMSERVR' newPass'/*'
'command c/GCS GCS/GCS' newPass'/*'
'command c/GCSXA GCSXA/GCSXA' newPass'/*'
'command c/SYSMAINT SYSMAINT/SYSMAINT' newPass'/*'
'command c/OPERATOR OPERATOR/OPERATOR' newPass'/*'
'command c/OP1 OP1/OP1' newPass'/*'
'command c/EREP EREP/EREP' newPass'/*'
'command c/OPERATNS OPERATNS/OPERATNS' newPass'/*'
'command c/AUTOLOG1 AUTOLOG1/AUTOLOG1' newPass'/*'
'command c/DISKACNT DISKACNT/DISKACNT' newPass'/*'
'command c/OPERSYMP OPERSYMP/OPERSYMP' newPass'/*'
'command c/VMUTIL VMUTIL/VMUTIL' newPass'/*'
'command c/SYSDUMP1 SYSDUMP1/SYSDUMP1' newPass'/*'
'command c/5684042J 5684042J/5684042J' newPass'/*'
'command c/4OSASF40 4OSASF40/4OSASF40' newPass'/*'
'command c/OSASF OSASF/OSASF' newPass'/*'
'command c/OSAMAINT OSAMAINT/OSAMAINT' newPass'/*'
'command c/OSADMIN1 OSADMIN1/OSADMIN1' newPass'/*'
'command c/OSADMIN2 OSADMIN2/OSADMIN2' newPass'/*'
'command c/OSADMIN3 OSADMIN3/OSADMIN3' newPass'/*'
'command c/P684096K P684096K/P684096K' newPass'/*'
'command c/RSCS RSCS/RSCS' newPass'/*'
'command c/XCHANGE XCHANGE/XCHANGE' newPass'/*'
'command c/RSCSDNS RSCSDNS/RSCSDNS' newPass'/*'
'command c/TCPMAINT TCPMAINT/TCPMAINT' newPass'/*'
'command c/5VMTCP20 5VMTCP20/5VMTCP20' newPass'/*'
'command c/TCPIP TCPIP/TCPIP' newPass'/*'
'command c/IMAP IMAP/IMAP' newPass'/*'
'command c/FTPSERVE FTPSERVE/FTPSERVE' newPass'/*'
'command c/SMTP SMTP/SMTP' newPass'/*'
'command c/NAMESRV NAMESRV/NAMESRV' newPass'/*'
'command c/REXECD REXECD/REXECD' newPass'/*'
'command c/RXAGENT1 RXAGENT1/RXAGENT1' newPass'/*'
'command c/X25IPI X25IPI/X25IPI' newPass'/*'
'command c/PORTMAP PORTMAP/PORTMAP' newPass'/*'
'command c/NDBPMGR NDBPMGR/NDBPMGR' newPass'/*'
```

```
'command c/NDBSRV01 NDBSRV01/NDBSRV01' newPass'/*'
'command c/SNMPQE SNMPQE/SNMPQE' newPass'/*'
'command c/SNMPD SNMPD/SNMPD' newPass'/*'
'command c/ROUTED ROUTED/ROUTED' newPass'/*'
'command c/LPSERVE LPSERVE/LPSERVE' newPass'/*'
'command c/SNALNKA SNALNKA/SNALNKA' newPass'/*'
'command c/VMNFS VMNFS/VMNFS' newPass'/*'
'command c/VMKERB VMKERB/VMKERB' newPass'/*'
'command c/ADMSERV ADMSERV/ADMSERV' newPass'/*'
'command c/UFTD    UFTD/UFTD' newPass'/*'
'command c/BOOTPD BOOTPD/BOOTPD' newPass'/*'
'command c/TFTPD TFTPD/TFTPD' newPass'/*'
'command c/DHCPD DHCPD/DHCPD' newPass'/*'
'command c/MPROUTE MPROUTE/MPROUTE' newPass'/*'
'command c/SSLSERV SSLSERV/SSLSERV' newPass'/*'
'command c/MONWRITE MONWRITE/MONWRITE' newPass'/*'
'command c/5VMDIR10 5VMDIR10/5VMDIR10' newPass'/*'
'command c/BLDNUC  BLDNUC/BLDNUC' newPass'/*'
'command c/BLDCMS  BLDCMS/BLDCMS' newPass'/*'
'command c/AUDITOR AUDITOR/AUDITOR' newPass'/*'
'command c/SYSMON  SYSMON/SYSMON' newPass'/*'
'command c/VMRMSVM VMRMSVM/VMRMSVM' newPass'/*'
'command c/VMRMADMN VMRMADMN/VMRMADMN' newPass'/*'
'command c/5767002P 5767002P/5767002P' newPass'/*'
'command c/RACFVM RACFVM/RACFVM' newPass'/*'
'command c/RACFSMF RACFSMF/RACFSMF' newPass'/*'
'command c/RACMAINT RACMAINT/RACMAINT' newPass'/*'
'command c/AUTOLOG2 AUTOLOG2/AUTOLOG2' newPass'/*'
'command c/IBMUSER IBMUSER/IBMUSER' newPass'/*'
'command c/SYSADMIN SYSADMIN/SYSADMIN' newPass'/*'
'command c/BLDRACF  BLDRACF/BLDRACF' newPass'/*'
'command c/5VMPTK20 5VMPTK20/5VMPTK20' newPass'/*'
'command c/PERFSVM  PERFSVM/PERFSVM' newPass'/*'
'command c/5VMHCD20 5VMHCD20/4VMHCD40' newPass'/*'
'command c/CBDIODSP CBDIODSP/CBDIODSP' newPass'/*'
'command c/VSMSERVE VSMSERVE/VSMSERVE' newPass'/*'
'command c/IMAPAUTH IMAPAUTH/IMAPAUTH' newPass'/*'
'command c/DTCVSW1  DTCVSW1/DTCVSW1 ' newPass'/*'
'command c/DTCVSW2  DTCVSW2/DTCVSW2 ' newPass'/*'
'command c/MIGMAINT MIGMAINT/MIGMAINT' newPass'/*'
'command c/LNXMAINT LNXMAINT/LNXMAINT' newPass'/*'
'command c/BLDSEG BLDSEG/BLDSEG' newPass'/*'

/* change mindisk passwords */
'command c/ALL      WRITE    MULTIPLE/ALL' newPass newPass'/*'
'command c/ALL      WTCPMAIN MTCPMAIN/ALL' newPass newPass'/*'
'command c/RADMSERV WADMSERV MADMSERV/'newPass newPass newPass'/*'
'command c/RAUDITOR WAUDITOR MAUDITOR/'newPass newPass newPass'/*'
'command c/RAUTOLOG WAUTOLOG MAUTOLOG/'newPass newPass newPass'/*'
'command c/RAVSOBJ  WAVSOBJ  MAVSOBJ/'newPass newPass newPass'/*'
'command c/RBATCH   WBATCH   MBATCH/'newPass newPass newPass'/*'
'command c/RBOOTPD  WBOOTPD  MBOOTPD/'newPass newPass newPass'/*'
'command c/RCATALOG WCATALOG/'newPass newPass'/*'
'command c/RCONTROL WCONTROL/'newPass newPass'/*'
'command c/RCRRLOG1 WCRRLOG1/'newPass newPass'/*'
'command c/RDATA    WDATA/'newPass newPass'/*'
'command c/RDHCPD   WDHCPD   MDHCPD/'newPass newPass newPass'/*'
'command c/RDVF     WDVF     MDVF/'newPass newPass newPass'/*'
'command c/READ     WRITE    MULTIPLE/'newPass newPass newPass'/*'
'command c/READ     WRITE/'newPass newPass'/*'
```

```
'command c/RFTPSERV WFTPSERV MFTPSERV/'newPass newPass newPass'/*'
'command c/RGCS     WGCS     MGCS/'newPass newPass newPass'/*'
'command c/RIMAP    WIMAP    MIMAP/'newPass newPass newPass'/*'
'command c/RLOG1    WLOG1/'newPass newPass newPass'/*'
'command c/RLOG2    WLOG2/'newPass newPass newPass'/*'
'command c/RLPSERVE WLPSERVE MLPSERVE/'newPass newPass newPass'/*'
'command c/RMAINT   WMAINT   MMAINT/'newPass newPass newPass'/*'
'command c/RMPROUTE WMPROUTE MMPROUTE/'newPass newPass newPass'/*'
'command c/RNAMESRV WNAMESRV MNAMESRV/'newPass newPass newPass'/*'
'command c/RNDBPMGR WNDBPMGR MNDBPMGR/'newPass newPass newPass'/*'
'command c/RNDBSRV0 WNDBSRV0 MNDBSRV0/'newPass newPass newPass'/*'
'command c/RPORTMAP WPORTMAP MPORTMAP/'newPass newPass newPass'/*'
'command c/RREXECD  WREXECD  MREXECD/'newPass newPass newPass'/*'
'command c/RROUTED  WROUTED  MROUTED/'newPass newPass newPass'/*'
'command c/RSERVER  WSERVER/'newPass newPass'/*'
'command c/RSMTP    WSMTP    MSMTP/'newPass newPass newPass'/*'
'command c/RSNALNKA WSNALNKA MSNALNKA/'newPass newPass newPass'/*'
'command c/RSNMPD   WSNMPD   MSNMPD/'newPass newPass newPass'/*'
'command c/RSNMPQE  WSNMPQE  MSNMPQE/'newPass newPass newPass'/*'
'command c/RSSLSERV WSSLSERV MSSLSERV/'newPass newPass newPass'/*'
'command c/RSYSMON  WSYSMON  MSYSMON/'newPass newPass newPass'/*'
'command c/RTCPIP   WTCPIP   MTCPIP/'newPass newPass newPass'/*'
'command c/RTCPMAIN WTCPMAIN MTCPMAIN/'newPass newPass newPass'/*'
'command c/RTFTPD   WTFTPD   MTFTPD/'newPass newPass newPass'/*'
'command c/RTSAFOBJ WTSAFOBJ MTSAFOBJ/'newPass newPass newPass'/*'
'command c/RUFTD    WUFTD    MUFTD/'newPass newPass newPass'/*'
'command c/RVMKERB  WVMKERB  MVMKERB/'newPass newPass newPass'/*'
'command c/RVMNFS   WVMNFS   MVMNFS/'newPass newPass newPass'/*'
'command c/RX25IPI  WX25IPI  MX25IPI/'newPass newPass newPass'/*'
'command c/R4TCPIP  W4TCPIP  M4TCPIP/'newPass newPass newPass'/*'
'command c/RDTCVSW1 WDTCVSW1 MDTCVSW1/'newPass newPass newPass'/*'
'command c/RDTCVSW2 WDTCVSW2 MDTCVSW2/'newPass newPass newPass'/*'
'command c/RCRRLOG2 WCRRLOG2/'newPass newPass'/*'
'command c/MR READ/'MR newPass'/*'
```

B.1.3 The LABEL520 EXEC

Following is the code for the EXEC that changes the system labels of a z/VM 5.2 system:

```
/*+------------------------------------------------------------------+*/
/*| EXEC: LABEL520  wrapper around CPFMTXA to LABEL and ALLOC DASD   |*/
/*|  retVal: 0 - success                                             |*/
/*|          1 - help was asked for or given                         |*/
/*|          2 - user is not sure                                    |*/
/*|          3 - DASD (minidisk) range is not valid                  |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT   |*/
/*+------------------------------------------------------------------+*/

/*------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
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 upper arg res spl pag w01 w02 .
if (w02 = '') then call help

/* Construct the two character label prefix */
firstChar = 'M' /* change this for an LPAR ID other than 'M' */
labelPrefix = firstChar'V'

/* Construct the 5 labels */
resLabel = getLabel(labelPrefix res)
splLabel = getLabel(labelPrefix spl)
pagLabel = getLabel(labelPrefix pag)
w01Label = getLabel(labelPrefix w01)
w02Label = getLabel(labelPrefix w02)

/* Ask "Are you sure?" */
say 'The volumes are:'
'CP Q' res spl pag w01 w02
say ''
say 'The system volume labels will become:'
say resLabel splLabel pagLabel w01Label w02Label
say ''
say 'ARE YOU SURE you want to relabel the DASD (y/n)?'
parse upper pull answer
ansFirstChar = substr(answer, 1, 1)
if (ansFirstChar ^= 'Y') then exit 2

/* Label the 4 volumes: RES is 123, W01 is 124, W02 is 125, SPL is 122 */
'CP TERM MORE 1 1'
'CPFMTXA 123' resLabel 'LABEL'
'CPFMTXA 124' w01Label 'LABEL'
'CPFMTXA 125' w02Label 'LABEL'
'CPFMTXA 122' splLabel 'LABEL'

/* LINK the 520PAG volume which is $PAGE$ A03, label it, DETACH it */
'CP LINK $PAGE$ A03 A03 MR'
'CPFMTXA A03' pagLabel 'LABEL'
'CP DET A03'
'CP TERM MORE 50 10'
exit

/*+------------------------------------------------------------------+*/
help: procedure
/*+------------------------------------------------------------------+*/
  parse source . . fn .
  say ""
  say "Synopsis:"
  say ""
  say "Relabel the five system volumes (520RES, 520W01, ...) to VV<xxxx>"
  say "   where <xxxx> is the 4 digit address"
  say ""
  say "Syntax is:"
  say ""
  say "   >>---LABEL520--res--spl--pag--w01--w02------------------------><"
  say ""
  say " where res, spl, pag, w01 and w02 are 4 digit virtual addresses"
  say " of the volumes that z/VM 5.2 is installed onto"
  say ""
exit 1
```

```
/*+------------------------------------------------------------------+*/
getLabel: procedure
/*| Compose the six character label of a minidisk                    |*/
/*|  parm 1: labelPrefix - first two characters of label             |*/
/*|  parm 2: disk - vaddr of length 1, 2, 3 or 4                     |*/
/*|  return: the 6 character label                                   |*/
/*+------------------------------------------------------------------+*/
  arg labelPrefix disk
  if (DATATYPE(disk, 'X') = 0) then
  do
    say "Error:" disk "is not a hexadecimal number"
    call help
  end
  diskLen = length(disk)
  select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
  end /* select */
return label /* from getLabel */
```

B.1.4 The LABEL520 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.2 `USER DIRECT` file:

```
/* LABEL520 - change '520xxx' labels in SYSTEM CONFIG or USER DIRECT */
/*-------------------------------------------------------------------
THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
------------------------------------------------------------------*/

firstChar = 'M' /* change this for an LPAR ID other than 'M' */
parse upper arg res spl pag w01 w02 .

parse upper arg fn ft fm '(' options ')' res spl pag w01 w02 .
if (w02 = '') then
do
  say "5 arguments required - exiting"
  exit
end
'command set stay on'
'command set num on'
'command set nulls on'
'command set serial off'
```

```
'command set cmdline bottom'
'command set curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'
'command top'
'command c/520RES/'firstChar'V'res'/*'
'command c/520W01/'firstChar'V'w01'/*'
'command c/520W02/'firstChar'V'w02'/*'
'command c/520SPL/'firstChar'V'spl'/*'
'command c/520PAG/'firstChar'V'pag'/*'
```

B.2 Linux souce code

This section lists source code that is common regardless of Linux distribution

B.2.1 Bash scripts on the controller

Following is the source code for the bash scripts that run on the controller

B.2.1.1 The nightly.sh script

Following is the content of the bash script /etc/cron.daily/nightly.sh. It runs scripts to backup important z/VM and Linux data.

```
!/bin/bash
# -----------------------------------------------------------------------------
# For details on this code see the books "z/VM and Linux on IBM System z:
# The Virtualization Cookbook for <x>" on the Web at: http://linuxvm.org/present
# where <x> is "SLES 10" or "RHEL 5"
#
# This code is shipped under the Common Public License. See the file
# Common-Public-License.txt or http://www.opensource.org/licenses/cpl1.0.txt
# -----------------------------------------------------------------------------
#
# nightly.sh - do the following on the controller each night
#   run backupVM.sh to backup VM files
#   run backupLinux.sh to backup /etc/ directories on Linux
# -----------------------------------------------------------------------------
/sbin/backupVM.sh
/sbin/backupLinux.sh
```

B.2.1.2 The backupVM.sh script

Following is the content of the bash script /etc/cron.daily/nightly.sh. It doesTHIS

```
#!/bin/bash
# -----------------------------------------------------------------------------
# For details on this code see the books "z/VM and Linux on IBM System z:
# The Virtualization Cookbook for <x>" on the Web at: http://linuxvm.org/present
# where <x> is "SLES 10" or "RHEL 5"
#
# This code is shipped under the Common Public License. See the file
# Common-Public-License.txt or http://www.opensource.org/licenses/cpl1.0.txt
# -----------------------------------------------------------------------------
#
# backupVM.sh  - do a backup of z/VM files by linking to important disks:
```

```
#  MAINT 191 - USER DISKMAP
#  MAINT 2CC - USER DIRECT
#  MAINT CF1 - SYSTEM CONFIG
#  TCPMAINT 198 - <systemID> TCPIP, SYSTEM DTCPARMS
#  TCPMAINT 592 - TCPIP DATA
#  AUTOLOG1 191 - PROFILE EXEC
#  LNXMAINT 192 - * PARMFILE, * PARM-S10, SLES* *
# The system that this is run on should have the LNKNOPAS directory option so
# all the above minidisks can be linked via vmcp with no password necessary
#
#+---------------------------------------------------------------------------+
function cpCmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg 1-n: the command to issue
#   Return: the command's return code
#+---------------------------------------------------------------------------+
 {
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  ret_val=0
  ret_val=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
  return $ret_val
 }

#+---------------------------------------------------------------------------+
function linkAccess()
# link and access a minidisk
#   Arg 1: the source disk user ID
#   Arg 2: the source disk virtual device number
#   Arg 3: placeholder for code readability
#   Arg 4: the virtual device number to link the minidisk as
#+---------------------------------------------------------------------------+
 {
  userID=$1
  vaddr=$2
  linkVaddr=$4
  cpCmd QUERY VIRTUAL $linkVaddr
  queryRC=$?
  if [ $queryRC = 40 ]; then # linkVaddr is free
    cpCmd LINK $userID $vaddr $linkVaddr RR
    rc=$?
    if [ $rc != 0 ]; then # can't link - punt
      echo "Error: unable to link $userID $vaddr read-only"
      echo "LINK $userID $vaddr $linkVaddr RR returned $rc"
      exit
    fi
  else # linkVaddr is not free
    echo "Error: Virtual address $linkVaddr is not free?"
    echo "QUERY VIRTUAL $linkVaddr returned $queryRC"
  fi
  # activate new disk then sleep 1 second for CP to catch up
  chccwdev -e $linkVaddr > /dev/null 2>&1
  sleep 1
 }

#+---------------------------------------------------------------------------+
function getFile()
# get a file on a CMS disk
#   Arg 1: the file or file pattern to get
#   Arg 2: placeholder for code readability
#   Arg 3: the virtual device number to detach
#+--------------------------------------------------------------------------+
 {
  file=$1
  vaddr=$3
  CMSdevice=`cat /proc/dasd/devices | grep $vaddr | awk '{print $7}'`
  cmsfscat -a -d /dev/$CMSdevice $file > $file
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: cmsfscat returned $rc backing up $file"
  else
    echo "Backing up $file seems to have succeeded"
  fi
 }

#+--------------------------------------------------------------------------+
function getFiles()
# get all files on a CMS disk
#   Arg 1: the virtual device number to copy files from
#+--------------------------------------------------------------------------+
 {
  vaddr=$1
  CMSdevice=`cat /proc/dasd/devices | grep $vaddr | awk '{print $7}'`
  let i=0
  cmsfslst -d /dev/$CMSdevice | while read oneline
  do # first record is headers, second is "DIRECTOR", third is "ALLOCMAP"
    if (($i > 2)); then # we are into the CMS files
      fn=`echo $oneline | awk '{print $1}'`
      ft=`echo $oneline | awk '{print $2}'`
      cmsfscat -a -d /dev/$CMSdevice $fn.$ft > $fn.$ft
      rc=$?
      if [ $rc != 0 ]; then # error
        echo "Error: cmsfscat returned $rc backing up $fn.$ft"
      else
        echo "Backing up $fn.$ft seems to have succeeded"
      fi
    fi
    let i=$i+1
  done
 }

#+--------------------------------------------------------------------------+
function relDetach()
# link and access a minidisk
#   Arg 1: the virtual device number to detach
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
  chccwdev -d $1 > /dev/null 2>&1
  cpCmd DET $1
 }

# main()
cd /backup/vm

# run the getVMinfo script to save some basic VM information
getVMinfo > zVMinfo.txt

# get USER DISKMAP from MAINT 191
linkAccess MAINT 191 as 1191
```

```
getFile USER.DISKMAP from 1191
relDetach 1191

# get USER DIRECT from MAINT 2CC
linkAccess MAINT 2CC as 12cc
getFile USER.DIRECT from 12cc
relDetach 12cc

# get SYSTEM CONFIG from MAINT CF1
linkAccess MAINT CF1 as 1cf1
getFile SYSTEM.CONFIG from 1cf1
relDetach 1cf1

# get TCPIP files from TCPMAINT 198
systemID=`vmcp q userid | awk '{print $3}'`
linkAccess TCPMAINT 198 as 1198
getFile $systemID.TCPIP from 1198
getFile SYSTEM.DTCPARMS from 1198
relDetach 1198

# get TCPIP DATA from TCPMAINT 592
linkAccess TCPMAINT 592 as 1592
getFile TCPIP.DATA from 1592
relDetach 1592

# get PROFILE EXEC from AUTOLOG1 191 - rename to different FT
linkAccess AUTOLOG1 191 as 2191
CMSdevice=`cat /proc/dasd/devices | grep 2191 | awk '{print $7}'`
cmsfscat -a -d /dev/$CMSdevice PROFILE.EXEC > PROFILE.EXECAUT1
relDetach 2191

# Get all files from LNXMAINT 192 which is our 191 disk
getFiles 191
```

B.2.1.3 The backupLinux.sh script

Following is the content of the bash script **/sbin/backupLinux.sh**. It loops through the directories under /backup/linux/ and updates each /etc/ directory via **rsync**:

```
#!/bin/bash
#
# ---------------------------------------------------------------------------
# For details on this code see the books "z/VM and Linux on IBM System z:
# The Virtualization Cookbook for <x>" on the Web at: http://linuxvm.org/present
# where <x> is "SLES 10" or "RHEL 5"
#
# This code is shipped under the Common Public License. See the file
# Common-Public-License.txt or http://www.opensource.org/licenses/cpl1.0.txt
# ---------------------------------------------------------------------------
#
BACKUP="/backup/linux"
cd $BACKUP
for i in *-on-*            # iterate through dirs with "-on-" in their names
do
  IP_addr=${i#LINUX*-on-}  # this chops the head off and grabs the IP address
  cd $BACKUP/$i            # change directory to next virtual server
  rsync -r --timeout=3 $IP_addr:/etc . > /dev/null 2>&1 # use rsync to backup
  rc=$?
  if [ $rc != 0 ]; then    # error
    echo "Error: rsync returned $rc while backing up $i"
  else
    echo "Backing up $i seems to have succeeded"
  fi
done
```

C

Source code specific to Linux

C.1 Obtaining and using the Web material

The PDF of this book is on the Internet at:

http://linuxvm.org/present/misc/virt-cookbook-RH5.pdf

The files associated with this book are in a gzipped tar file at:

http://linuxvm.org/present/misc/virt-cookbook-RH5.tgz

Download the tar file to your NFS server and use it as is described in section 6.1. After untarring the file, you will have a directory named `virt-cookbook-RH5/` for RHEL 5, or `virt-cookbook-S10/` for SLES 10.

This section lists the source code specific to RHEL 5 which is only the **`clone.sh`** script.

C.2 The clone.sh script

This section lists the **`clone.sh`** script:

```
#!/bin/bash
#
# clone.sh <LinuxUserID> - clone a Linux server running under z/VM
#
# For details on how this script works see the book:
#   "z/VM and Linux on IBM System z: The Virtualization Cookbook"
#   on the Web at: http://linuxvm.org/present/misc/virt-cookbook-1.pdf
#
# -----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# ----------------------------------------------------------------------------

#+--------------------------------------------------------------------------+
function help()
# give help
#+--------------------------------------------------------------------------+
 {
  echo "Usage: clone [options] target_linux_ID"
  echo ""
  echo "  Clone Linux system from minidisk $sdisk to target_linux_ID"
  echo "  options:"
  echo "    -v or --verbose: verbose"
  exit 1
 }

#+--------------------------------------------------------------------------+
function process_arguments()
# give help
#+--------------------------------------------------------------------------+
 {
  verbose="off"
  if [ $# = 0 ]; then # user did not pass an argument
    echo "error: at least one argument is needed"
    echo ""
    help
  fi
  for arg in $*
  do
    case $arg in
      -v|--verbose)
         verbose="on"
         ;;
      *)
        clone_id=`echo $arg | tr '[a-z]' '[A-Z]'`     # fold target user ID to upper
case
        ;;
    esac
  done
  if [ $clone_id = "none" ]; then # new linux user ID was not passed
    echo "Error: new Linux user ID must be an argument"
    echo ""
    help
  fi
 }

#+--------------------------------------------------------------------------+
function cp_cmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
  echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  ret_val=0
  ret_val=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
  return $ret_val
 }

#+--------------------------------------------------------------------------+
function check_target_id()
# Verify user ID exists and is logged off
#+--------------------------------------------------------------------------+
 {
  cp_cmd QUERY $clone_id
  case $? in
    0)  # user ID is logged on or disconnected
      echo "$clone_id user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$clone_id user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "$clone_id user ID must exist and be logged off"
      exit 4
  esac
 }

#+--------------------------------------------------------------------------+
function prepare_ipaddr()
# Prepare an IP address by adding a backslash before any "."s
# to make it 4 digits
#    Arg 1: The IP address to be modified
# Return:
#    The new value is written to the global variable new_ipaddr
#+--------------------------------------------------------------------------+
 {
  new_ipaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }

#+--------------------------------------------------------------------------+
function prepare_vaddr()
# Prepare a z/VM virtual address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable new_vaddr
#+--------------------------------------------------------------------------+
 {
  new_vaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    new_vaddr="0$new_vaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function copy_disk()
# Try to use z/VM FLASHCOPY to copy one disk to another.
#   Arg 1: Source minidisk vaddr
#   Arg 2: the word "to"
#   Arg 3: Target user ID
#   Arg 4: Target virtual address
# Return code
#   0: success with FLASHCOPY
#   1: success with dasdfmt and dd
#   2: Target vaddr already in use?
#   3: CP LINK failed
#   4: unable to enable target minidisk
#   5: can't find source disk in /dev/dasd/devices
#   6: can't find target disk in /dev/dasd/devices
#   7: dasdfmt failed
#   8: dd failed
#+--------------------------------------------------------------------------+
 {
  ret_val=0
  source_mdisk=$1
  target_userid=$3
  target_vaddr=$4
  cp_cmd QUERY VIRTUAL $target_vaddr
  rc=$?
  if [ $rc != 40 ]; then
    echo "Error: CP QUERY VIRTUAL $target_vaddr failed with $rc"
    echo "The target device address is already in use"
    return 2
  fi
  cp_cmd LINK $target_userid $source_mdisk $target_vaddr MR
  rc=$?
  if [ $rc != 0 ]; then # LINK failed
    echo "Error: CP LINK $target_userid $source_mdisk $target_vaddr failed with $rc"
    return 3
  fi
  cp_cmd FLASHCOPY $source_mdisk 0 END $target_vaddr 0 END
  rc=$?
  if [ $rc != 0 ]; then # FLASHCOPY failed
    echo "FLASHCOPY $source_mdisk $target_vaddr failed with $rc"
    echo "Falling back to dasdfmt and dd copy"
#   enable target disk
    sleep 1
    chccwdev -e $target_vaddr
    rc=$?
    if [ $rc != 0 ]; then # unable to enable target disk
      echo "Error: unable to enable $target_disk, rc from chccwdev = $rc"
      cp_cmd DETACH $target_vaddr
      return 4
    fi
#   get device name of source disk
    source_dev=`cat /proc/dasd/devices | grep "$source_mdisk(ECKD)" | awk '{ print $7
}'`
    if [ "$source_dev" = "" ]; then
      cat /proc/dasd/devices
      echo "Error: can't find $source_mdisk(ECKD) in /proc/dasd/devices"
      chccwdev -d $target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 5
    fi
#   get device name of target disk
    target_dev=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7
}'`
```

```
    if [ "$target_dev" = "" ]; then
      cat /proc/dasd/devices
      echo "Error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
      chccwdev -d $target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 6
    fi
#   dasdfmt target disk
    echo "Invoking command: dasdfmt -b 4096 -y -f /dev/$target_dev"
    dasdfmt -b 4096 -y -f /dev/$target_dev
    rc=$?
    if [ $rc != 0 ]; then # dasdfmt failed
      echo "Error: dasdfmt -b 4096 -y -f /dev/$target_dev failed with $rc"
      chccwdev -d $target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 7
    fi
#   copy from source disk to target via dd
    echo "Invoking command: dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev"
    dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev
    rc=$?
    if [ $rc != 0 ]; then # dd failed
      echo "Error: dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev failed with $rc"
      chccwdev -d $target_vaddr # clean up
      cp_cmd DETACH $target_vaddr
      return 8
    fi
    chccwdev -d $target_vaddr
    echo "Copying disk via dasdfmt/dd succeeded ..."
    ret_val=1 # success with dasdfmt and dd
  fi # if FLASHCOPY failed
  sync # sync disks
  cp_cmd DETACH $target_vaddr
  echo "Copying disk via FLASHCOPY succeeded ..."
  return $ret_val
 } # copy_disk()

#+---------------------------------------------------------------------------+
function get_conffile_info()
# Bring 191 minidisk online to be read by cmsfs and check for two CONF-RH5 files
#+---------------------------------------------------------------------------+
 {
  # recycle 191 to pick up latest changes
  chccwdev -d 191
  chccwdev -e 191
#  sleep 1 # timing problem?
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "unable to enable 191, rc from chccwdev = $rc"
    exit 5
  fi
  CMSdisk=`lsdasd | grep 0191 | awk '{ print $7 }'`
  cmsfslst -d /dev/$CMSdisk | grep -i $clone_id | grep CONF-RH5
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $clone_id CONF-RH5 not found on 191 minidisk. Exiting"
    exit 6
  fi

# get informaton about target
```

```
  export local $(cmsfscat -a -d /dev/$CMSdisk $clone_id.CONF-RH5)
  target_hostname=$HOSTNAME
  target_domain=${HOSTNAME#*.}
  target_host=${HOSTNAME%.$target_domain}
  target_IP=$IPADDR
  target_DNS=$DNS
  target_GW=$GATEWAY
  target_netmask=$NETMASK
  target_network=$NETWORK
  target_broadcast=$BROADCAST
  prepare_vaddr $SUBCHANNELS
  target_subchannels=$new_vaddr

# get information about source
  source_guestID=`cat /proc/sysinfo | grep "VM00 Name" | awk '{ print $3 }'`
  export local $(cmsfscat -a -d /dev/$CMSdisk $source_guestID.CONF-RH5)
  source_hostname=$HOSTNAME
  source_domain=${HOSTNAME#*.}
  source_host=${HOSTNAME%.$source_domain}
  prepare_ipaddr $IPADDR
  source_IP=$new_ipaddr
  prepare_ipaddr $DNS
  source_DNS=$new_ipaddr
  prepare_ipaddr $GATEWAY
  source_GW=$new_ipaddr
  prepare_ipaddr $NETMASK
  source_netmask=$new_ipaddr
  prepare_ipaddr $NETWORK
  source_network=$new_ipaddr
  prepare_ipaddr $BROADCAST
  source_broadcast=$new_ipaddr
  prepare_vaddr $SUBCHANNELS
  source_subchannels=$new_vaddr
 }

#+--------------------------------------------------------------------------+
function ask_are_you_sure()
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: this will copy disk $sdisk to $clone_id 100"
  echo "New host name will be:  $target_hostname"
  echo "New TCP/IP address will be: $target_IP"
  echo "Other network data is retrieved from $clone_id CONF-RH5 on 191 disk"
  echo -n "Are you sure you want to overwrite these disks (y/n): "
  read ans
  if [ $ans != "y" ]; then
    exit 7
  fi
  echo "Please enter a brief description of this server"
  read descr
 }

#+--------------------------------------------------------------------------+
function copy_system()
# copy master image on $sdisk (default 100) to target user ID
#+--------------------------------------------------------------------------+
 {
  echo "Copying $sdisk root file system to $clone_id ..."
  copy_disk $sdisk to $clone_id 1100 # copy source to target root file system
  rc=$?
  if [ $rc -gt 1 ]; then # both FLASHCOPY and dasdfmt/dd failed
    echo "Copying disk failed with $rc"
    exit 9
  fi
 }

#+--------------------------------------------------------------------------+
function modify_cloned_image()
# Mount newly copied system over /mnt/rhel5cloned and modify networking info
#   Arg 1: target userid
#   Arg 2: target minidisk
#   Arg 3: target virtual device address
# Return code
#   0: success
#+--------------------------------------------------------------------------+
 {
  target_userid=$1
  target_mdisk=$2
  target_vaddr=$3
  echo "Mounting newly cloned image over $cloned_mnt_pt ..."
  cp_cmd LINK $target_userid $target_mdisk $target_vaddr MR
  rc=$?
  sleep 1
  if [ $rc != 0 ]; then # LINK failed
    echo "Fatal error: CP LINK $target_userid $target_mdisk $target_vaddr MR failed with
$rc"
    cp_cmd DET $target_vaddr
    exit 10
  fi
  if [ ! -d $cloned_mnt_pt ]; then
    mkdir $cloned_mnt_pt
    rc=$?
    if [ $rc != 0 ]; then
      echo "Fatal Error: mkdir $cloned_mnt_pt failed with $rc"
      cp_cmd DET $target_vaddr
      exit 10
    fi
  fi
  chccwdev -e $target_vaddr
  sleep 2
  rc=$?
  if [ $rc != 0 ]; then
    echo "Fatal error: chccwdev -e $target_vaddr failed with $rc"
    cp_cmd DET $target_vaddr
    exit 10
  fi
  cloned_DASD=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7 }'`
  if [ "$cloned_DASD" = "" ]; then
    cat /proc/dasd/devices
    echo "Fatal error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
    chccwdev -d $target_vaddr
    exit 10
  fi
  cloned_fs="/dev/${cloned_DASD}1"
  echo "Mounting $cloned_fs over $cloned_mnt_pt ..."
  mount $cloned_fs $cloned_mnt_pt
  rc=$?
  if [ $rc != 0 ]; then
```

```
    echo "Fatal error: mount $cloned_fs $cloned_mnt_pt failed with $rc"
    lsdasd
    cp_cmd DET $target_vaddr
    exit 10
  fi
  echo "Modifying cloned image under $cloned_mnt_pt ..."
  sed --in-place -e "s/$source_IP/$target_IP/g" \
    -e "s/$source_hostname/$target_hostname/g" \
    -e "s/$source_host/$target_host/g" $cloned_mnt_pt/etc/hosts
  sed --in-place -e "s/$source_hostname/$target_hostname/g" \
    -e "s/$source_GW/$target_GW/g" $cloned_mnt_pt/etc/sysconfig/network
  sed --in-place -e "s/$source_IP/$target_IP/g" \
    -e "s/$source_netmask/$target_netmask/g" \
    -e "s/$source_broadcast/$target_broadcast/g" \
    -e "s/$source_network/$target_network/g" \
    -e "s/$source_subchannels/$target_subchannels/g" \
    $cloned_mnt_pt/etc/sysconfig/network-scripts/ifcfg-eth0
  sed --in-place -e "s/$source_domain/$target_domain/g" \
    -e "s/$source_DNS/$target_DNS/g" $cloned_mnt_pt/etc/resolv.conf

# Regenerate SSH keys
  echo "Regenerating SSH keys in $cloned_mnt_pt/etc/ssh/ ..."
  rm $cloned_mnt_pt/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_key

# Remove any old entry, then copy clone's public key to known_hosts file
  echo "Adding $target_IP to known_hosts file"
  cd /root/.ssh
  if [ -f known_hosts ]; then
    grep -v $target_IP known_hosts > known_hosts.temp
    mv known_hosts.temp known_hosts
  fi

# construct known_hosts entry with three fields: IP@ "ssh-rsa" clones_public_key
  new_key=`cat $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key.pub | awk '{ print $2 }'`
  echo "$target_IP ssh-rsa $new_key" >> known_hosts

# clean up
  sync # sync disks
  umount $cloned_mnt_pt
  chccwdev -d $target_vaddr
  cp_cmd DETACH $target_vaddr
  return 0
 }

#+--------------------------------------------------------------------------+
function make_backup_dir()
# Create a directory /backup/<VM-ID>-on-<IP.addr> if it doesn't exist
# e.g. LINUX04 with IP@ 129.40.178.44 would create /backup/LINUX04-on-129.40.178.44/
#+--------------------------------------------------------------------------+
 {
  if [ -d $backup_dir ]; then # main backup directory exists
    echo "Creating a directory under $backup_dir"
    cd $backup_dir
    backup_subdir=$clone_id-on-$target_IP
    if [ ! -d $backup_subdir ]; then # backup subdirectory does not exist
      mkdir $backup_subdir
      if [ ! $? = 0 ]; then # there was an error from mkdir command
```

```
        echo "Error creating directory $backup_dir/$backup_subdir/"
      else
        echo "Created directory $backup_dir/$backup_subdir/"
        echo $descr > $backup_subdir/descLog.txt
      fi
    else # directory exists - create descLog.txt file
      echo $descr > $backup_subdir/descLog.txt
    fi
  fi
 }

# main() # some important global variables
sdisk=0100                                 # source disk
clone_id="none"                            # target user ID
was="no"
db2="no"
mqs="no"
umask 002                                  # new dir/files set group write bit
cloned_mnt_pt="/mnt/rhel5cloned"           # directory of temporary mount point
backup_dir="/backup/linux"                 # directory of Linux backups

process_arguments $@                       # process arguments passed by user
if [ $verbose = "on" ]; then set -vx; fi   # turn debug on
check_target_id                            # verify user ID to clone to
get_conffile_info                          # info about source and target
ask_are_you_sure                           # confirm disk overwrite
copy_system                                # copy disk $sdisk to target ID
modify_cloned_image $clone_id $sdisk 1100  # modify newly copied system
cp_cmd XAUTOLOG $clone_id                  # bring new clone to life
make_backup_dir                            # make a backup directory
if [ $verbose = "on" ]; then set +vx; fi   # turn debug off
echo "Successfully cloned minidisk $sdisk to $clone_id"
echo "You should be able to ping $target_IP within one minute"
exit 0
```

Index